

Monopoly

Generat per Doxygen 1.8.20

1 Projecte de Programació (GEINF/GEB - UdG)	1
1.1 Primavera 2021	1
2 Índex Jeràrquic	3
2.1 Jerarquia de Classes	3
3 Índex de Classes	5
3.1 Llista de Classes	5
4 Índex de Fitxers	7
4.1 Llista dels Fitxers	7
5 Documentació de les Classes	9
5.1 Referència de la Classe Board	9
5.1.1 Descripció Detallada	10
5.1.2 Documentació del Constructor i el Destructor	10
5.1.2.1 Board()	11
5.1.3 Documentació de les Funcions Membre	11
5.1.3.1 addBox()	11
5.1.3.2 addPlayer()	12
5.1.3.3 getBox()	13
5.1.3.4 getSize()	14
5.1.3.5 haveAvailableFields()	14
5.1.3.6 haveOwner()	15
5.1.3.7 isBankrupt()	16
5.1.3.8 movePlayer()	17
5.1.3.9 numberOfAgrupationField()	19
5.1.3.10 randomField()	20
5.1.3.11 selectOption()	20
5.1.3.12 setBoxesNr()	21
5.1.3.13 toString()	22
5.1.3.14 transferProperties()	23
5.1.3.15 tryToRunCardToAlive()	24
5.1.3.16 tryToSellToAlive()	25
5.1.4 Documentació de les Dades Membre	26
5.1.4.1 board	26
5.1.4.2 boxes_nr	27
5.1.4.3 players	27
5.2 Referència de la Classe Box	27
5.2.1 Descripció Detallada	28
5.2.2 Documentació del Constructor i el Destructor	28
5.2.2.1 Box()	28
5.2.3 Documentació de les Funcions Membre	29
5.2.3.1 getPosition()	29

5.2.3.2 getType()	30
5.2.3.3 playerInPosition()	30
5.2.3.4 print()	31
5.2.4 Documentació de les Dades Membre	32
5.2.4.1 name	32
5.2.4.2 position	32
5.2.4.3 type	32
5.3 Referència de la Classe BoxBet	33
5.3.1 Descripció Detallada	33
5.3.2 Documentació del Constructor i el Destructor	34
5.3.2.1 BoxBet()	34
5.3.3 Documentació de les Funcions Membre	34
5.3.3.1 betResult()	34
5.3.4 Documentació de les Dades Membre	35
5.3.4.1 combinations	35
5.4 Referència de la Classe BoxDirectCommand	35
5.4.1 Descripció Detallada	36
5.4.2 Documentació del Constructor i el Destructor	36
5.4.2.1 BoxDirectCommand()	37
5.4.3 Documentació de les Funcions Membre	37
5.4.3.1 getCard()	37
5.4.4 Documentació de les Dades Membre	38
5.4.4.1 function	38
5.5 Referència de la Classe BoxEmpty	38
5.5.1 Descripció Detallada	39
5.5.2 Documentació del Constructor i el Destructor	39
5.5.2.1 BoxEmpty()	39
5.6 Referència de la Classe BoxField	40
5.6.1 Descripció Detallada	42
5.6.2 Documentació del Constructor i el Destructor	42
5.6.2.1 BoxField()	42
5.6.3 Documentació de les Funcions Membre	43
5.6.3.1 build()	43
5.6.3.2 buy()	43
5.6.3.3 getGroup()	44
5.6.3.4 getName()	45
5.6.3.5 getNumberOfApartaments()	45
5.6.3.6 getNumberOfHotels()	46
5.6.3.7 getOwner()	47
5.6.3.8 getPrice()	47
5.6.3.9 getRent()	48
5.6.3.10 hotelBuildable()	49

5.6.3.11 houseBuildable()	49
5.6.3.12 houseBuildableType()	50
5.6.3.13 isBought()	50
5.6.3.14 numberOfWorkBuildable()	51
5.6.3.15 priceToBuild()	51
5.6.3.16 sell()	52
5.6.3.17 toString()	53
5.6.4 Documentació de les Dades Membre	53
5.6.4.1 basic_rent	53
5.6.4.2 bought	53
5.6.4.3 buildable	54
5.6.4.4 builded	54
5.6.4.5 building_price	54
5.6.4.6 buildings_rent	54
5.6.4.7 group	54
5.6.4.8 group_rent	55
5.6.4.9 hotel	55
5.6.4.10 hotel_price	55
5.6.4.11 hotel_rent	55
5.6.4.12 max_buildings	55
5.6.4.13 name	56
5.6.4.14 owner	56
5.6.4.15 price	56
5.7 Referència de la Classe BoxStart	56
5.7.1 Descripció Detallada	57
5.7.2 Documentació del Constructor i el Destructor	57
5.7.2.1 BoxStart()	57
5.7.3 Documentació de les Funcions Membre	58
5.7.3.1 getType()	58
5.8 Referència de la Classe Card	58
5.8.1 Descripció Detallada	59
5.8.2 Documentació del Constructor i el Destructor	59
5.8.2.1 Card()	59
5.8.3 Documentació de les Funcions Membre	60
5.8.3.1 getType()	60
5.8.3.2 isPostposable()	61
5.8.4 Documentació de les Dades Membre	61
5.8.4.1 postposable	61
5.8.4.2 type	61
5.9 Referència de la Classe CardCharge	62
5.9.1 Descripció Detallada	63
5.9.2 Documentació del Constructor i el Destructor	63

5.9.2.1 CardCharge()	63
5.9.3 Documentació de les Funcions Membre	63
5.9.3.1 execute()	64
5.9.3.2 getQuantity()	64
5.9.3.3 toString()	65
5.9.4 Documentació de les Dades Membre	65
5.9.4.1 message	66
5.9.4.2 messages	66
5.9.4.3 quantity	66
5.10 Referència de la Classe CardFine	66
5.10.1 Descripció Detallada	67
5.10.2 Documentació del Constructor i el Destructor	67
5.10.2.1 CardFine()	68
5.10.3 Documentació de les Funcions Membre	68
5.10.3.1 execute()	68
5.10.3.2 toString()	69
5.10.4 Documentació de les Dades Membre	70
5.10.4.1 message	70
5.10.4.2 messages	70
5.10.4.3 quantity	70
5.11 Referència de la Classe CardGet	71
5.11.1 Descripció Detallada	72
5.11.2 Documentació del Constructor i el Destructor	72
5.11.2.1 CardGet()	72
5.11.3 Documentació de les Funcions Membre	72
5.11.3.1 execute()	72
5.11.3.2 fieldSelect()	73
5.11.3.3 playerSelect()	74
5.11.3.4 toString()	75
5.12 Referència de la Classe CardGive	76
5.12.1 Descripció Detallada	77
5.12.2 Documentació del Constructor i el Destructor	77
5.12.2.1 CardGive()	77
5.12.3 Documentació de les Funcions Membre	78
5.12.3.1 execute()	78
5.12.3.2 fieldSelect()	79
5.12.3.3 playerSelect()	80
5.12.3.4 toString()	81
5.13 Referència de la Classe CardGo	81
5.13.1 Descripció Detallada	82
5.13.2 Documentació del Constructor i el Destructor	82
5.13.2.1 CardGo()	82

5.13.3 Documentació de les Funcions Membre	83
5.13.3.1 execute()	83
5.13.3.2 toString()	84
5.13.4 Documentació de les Dades Membre	85
5.13.4.1 position	85
5.14 Referència de la Classe CardPay	85
5.14.1 Descripció Detallada	86
5.14.2 Documentació del Constructor i el Destructor	86
5.14.2.1 CardPay()	86
5.14.3 Documentació de les Funcions Membre	87
5.14.3.1 execute()	87
5.14.3.2 toString()	88
5.14.4 Documentació de les Dades Membre	89
5.14.4.1 quantity	89
5.15 Referència de la Classe CPUPlayer	89
5.15.1 Descripció Detallada	91
5.15.2 Documentació del Constructor i el Destructor	91
5.15.2.1 CPUPlayer()	92
5.15.3 Documentació de les Funcions Membre	92
5.15.3.1 bankruptcy()	92
5.15.3.2 betQuantity()	93
5.15.3.3 betValue()	94
5.15.3.4 build()	95
5.15.3.5 buildApartment()	96
5.15.3.6 buildChoice()	97
5.15.3.7 buy()	98
5.15.3.8 buyBuyerOffer()	99
5.15.3.9 buyFieldSelect()	100
5.15.3.10 buyInitialOffer()	101
5.15.3.11 buyPlayerSelect()	102
5.15.3.12 buySellerOffer()	102
5.15.3.13 cardGetFieldSelect()	103
5.15.3.14 cardGetPlayerSelect()	104
5.15.3.15 cardGiveFieldSelect()	105
5.15.3.16 cardGivePlayerSelect()	106
5.15.3.17 cardPayPlayerSelect()	107
5.15.3.18 loanInitialOffer()	107
5.15.3.19 loanInterestOffer()	108
5.15.3.20 loanPlayerSelect()	109
5.15.3.21 loanTurnsOffer()	110
5.15.3.22 optionalActionSelector()	110
5.15.3.23 optionSelection()	112

5.15.3.24 postposableLuckCardChoice()	113
5.15.3.25 sellBuyerOffer()	114
5.15.3.26 sellFieldSelect()	115
5.15.3.27 sellInitialOffer()	116
5.15.3.28 start()	117
5.15.3.29 stringValueSelection()	118
5.15.4 Documentació de les Dades Membre	119
5.15.4.1 rand	119
5.16 Referència de la Classe JocMonopoly	119
5.16.1 Descripció Detallada	120
5.16.2 Documentació de les Funcions Membre	120
5.16.2.1 main()	120
5.16.2.2 printHelp()	121
5.17 Referència de la Classe JSONManager	122
5.17.1 Descripció Detallada	123
5.17.2 Documentació del Constructor i el Destructor	123
5.17.2.1 JSONManager()	123
5.17.3 Documentació de les Funcions Membre	124
5.17.3.1 readBoard()	124
5.17.3.2 readFile()	125
5.17.3.3 readRules()	126
5.17.4 Documentació de les Dades Membre	126
5.17.4.1 board_file	126
5.17.4.2 cards	126
5.17.4.3 initial_money	127
5.17.4.4 mode	127
5.17.4.5 rules_file	127
5.17.4.6 star_rewards	127
5.18 Referència de la Classe Monopoly	128
5.18.1 Descripció Detallada	129
5.18.2 Documentació del Constructor i el Destructor	129
5.18.2.1 Monopoly()	130
5.18.3 Documentació de les Funcions Membre	130
5.18.3.1 askInfoOfPlayers()	130
5.18.3.2 askNrOfPlayers()	131
5.18.3.3 checkEndGame()	132
5.18.3.4 endGame()	133
5.18.3.5 getCurrentBox()	134
5.18.3.6 getMode()	135
5.18.3.7 movePlayer()	135
5.18.3.8 nextPlayer()	136
5.18.3.9 play()	137

5.18.3.10 printPlayers()	139
5.18.3.11 startGame()	139
5.18.3.12 throwDice()	140
5.18.4 Documentació de les Dades Membre	141
5.18.4.1 board	141
5.18.4.2 cards	142
5.18.4.3 current_player	142
5.18.4.4 current_player_iterator	142
5.18.4.5 dice_result	142
5.18.4.6 initial_money	142
5.18.4.7 mode	143
5.18.4.8 optional_actions	143
5.18.4.9 players	143
5.18.4.10 start_rewards	143
5.19 Referència de la Classe Movement	144
5.19.1 Descripció Detallada	145
5.19.2 Documentació del Constructor i el Destructor	145
5.19.2.1 Movement()	145
5.19.3 Documentació de les Funcions Membre	146
5.19.3.1 betAction()	146
5.19.3.2 build()	147
5.19.3.3 buildApartament()	148
5.19.3.4 buildHotel()	149
5.19.3.5 buyField()	150
5.19.3.6 directComand()	151
5.19.3.7 fieldAction()	153
5.19.3.8 getCards()	154
5.19.3.9 getOutput()	154
5.19.3.10 luckAction()	155
5.19.3.11 optionalActions()	156
5.19.3.12 payRent()	157
5.19.3.13 runCard()	159
5.19.3.14 startAction()	160
5.19.4 Documentació de les Dades Membre	161
5.19.4.1 active_player	161
5.19.4.2 cards	161
5.19.4.3 current_box	161
5.19.4.4 output	161
5.19.4.5 start_rewards	161
5.20 Referència de la Classe OpActBuy	162
5.20.1 Descripció Detallada	163
5.20.2 Documentació del Constructor i el Destructor	163

5.20.2.1 OpActBuy()	163
5.20.3 Documentació de les Funcions Membre	163
5.20.3.1 buy()	163
5.20.3.2 buyNegociation()	165
5.20.3.3 execute()	166
5.20.3.4 fieldSelect()	167
5.20.3.5 playerSelect()	168
5.20.3.6 startOffer()	169
5.20.3.7 toString()	171
5.21 Referència de la Classe OpActLoan	171
5.21.1 Descripció Detallada	172
5.21.2 Documentació del Constructor i el Destructor	172
5.21.2.1 OpActLoan()	172
5.21.3 Documentació de les Funcions Membre	172
5.21.3.1 execute()	173
5.21.3.2 giveLoan()	173
5.21.3.3 negociation()	175
5.21.3.4 offer()	176
5.21.3.5 playerSelect()	177
5.21.3.6 toString()	178
5.22 Referència de la Classe OpActLuckCard	179
5.22.1 Descripció Detallada	180
5.22.2 Documentació del Constructor i el Destructor	180
5.22.2.1 OpActLuckCard()	180
5.22.3 Documentació de les Funcions Membre	180
5.22.3.1 execute()	180
5.22.3.2 toString()	181
5.23 Referència de la Classe OpActSell	182
5.23.1 Descripció Detallada	183
5.23.2 Documentació del Constructor i el Destructor	183
5.23.2.1 OpActSell()	183
5.23.3 Documentació de les Funcions Membre	183
5.23.3.1 doAuction()	184
5.23.3.2 execute()	185
5.23.3.3 selectFieldToSell()	186
5.23.3.4 selectPriceOfField()	187
5.23.3.5 toString()	188
5.24 Referència de la Classe optionalActions	189
5.24.1 Descripció Detallada	189
5.24.2 Documentació de les Funcions Membre	190
5.24.2.1 execute()	190
5.24.2.2 toString()	190

5.25 Referència de la Classe OutputManager	191
5.25.1 Descripció Detallada	191
5.25.2 Documentació del Constructor i el Destructor	191
5.25.2.1 OutputManager()	192
5.25.3 Documentació de les Funcions Membre	192
5.25.3.1 fileWrite()	192
5.25.4 Documentació de les Dades Membre	193
5.25.4.1 dev_file	193
5.26 Referència de la Classe Player	193
5.26.1 Descripció Detallada	196
5.26.2 Documentació del Constructor i el Destructor	196
5.26.2.1 Player()	196
5.26.3 Documentació de les Funcions Membre	196
5.26.3.1 addBox()	196
5.26.3.2 addLoan()	197
5.26.3.3 addLuckCard()	198
5.26.3.4 charge()	199
5.26.3.5 getBankruptcy()	199
5.26.3.6 getFields()	200
5.26.3.7 getLoans()	201
5.26.3.8 getLuckCards()	201
5.26.3.9 getMoney()	202
5.26.3.10 getName()	203
5.26.3.11 getPosition()	204
5.26.3.12 getType()	204
5.26.3.13 goToBankruptcy()	205
5.26.3.14 haveFields()	205
5.26.3.15 movePlayer()	206
5.26.3.16 numberOfAgrupationField()	206
5.26.3.17 numberOfBirthBuildingsAffordable()	207
5.26.3.18 optionSelection()	208
5.26.3.19 pay()	208
5.26.3.20 payLoans()	209
5.26.3.21 removeBox()	211
5.26.3.22 removeLuckCard()	211
5.26.3.23 stringValueSelection()	212
5.26.3.24 toString()	213
5.26.4 Documentació de les Dades Membre	213
5.26.4.1 bankruptcy	213
5.26.4.2 boxes_in_property	214
5.26.4.3 loans	214
5.26.4.4 luckCards	214

5.26.4.5 money	214
5.26.4.6 name	214
5.26.4.7 position	215
5.26.4.8 type	215
5.27 Referència de la Classe PlayerLoan	215
5.27.1 Descripció Detallada	216
5.27.2 Documentació del Constructor i el Destructor	216
5.27.2.1 PlayerLoan()	216
5.27.3 Documentació de les Funcions Membre	217
5.27.3.1 getLoaner()	217
5.27.3.2 isEnd()	217
5.27.3.3 nextTurn()	218
5.27.3.4 payLoan()	219
5.27.3.5 returnValue()	220
5.27.3.6 smallPrint()	220
5.27.3.7 toString()	221
5.28 Referència de la Classe TerminalPlayer	222
5.28.1 Descripció Detallada	222
5.28.2 Documentació del Constructor i el Destructor	223
5.28.2.1 TerminalPlayer()	223
5.28.3 Documentació de les Funcions Membre	223
5.28.3.1 optionSelection()	223
5.28.3.2 stringValueSelection()	224
5.29 Referència de la Classe UIFileSelector	224
5.29.1 Descripció Detallada	225
5.29.2 Documentació del Constructor i el Destructor	225
5.29.2.1 UIFileSelector()	226
5.29.3 Documentació de les Funcions Membre	226
5.29.3.1 createBrowseButtonAction()	226
5.29.3.2 createLabels()	227
5.29.3.3 createNextButtonAction()	228
5.29.3.4 defineButtons()	228
5.29.3.5 defineTextAreas()	229
5.29.3.6 defineTextFields()	229
5.29.3.7 getBoardFileName()	230
5.29.3.8 getRulesFileName()	230
5.29.3.9 getStatus()	231
5.29.3.10 startComponents()	231
5.29.4 Documentació de les Dades Membre	232
5.29.4.1 board_name	232
5.29.4.2 rules_name	233
5.29.4.3 status	233

6 Documentació dels Fitxers	235
6.1 Referència del Fitxer Board.java	235
6.1.1 Descripció Detallada	235
6.2 Referència del Fitxer Box.java	235
6.2.1 Descripció Detallada	235
6.3 Referència del Fitxer BoxBet.java	235
6.4 Referència del Fitxer BoxDirectCommand.java	236
6.5 Referència del Fitxer BoxEmpty.java	236
6.6 Referència del Fitxer BoxField.java	236
6.7 Referència del Fitxer BoxStart.java	236
6.8 Referència del Fitxer Card.java	236
6.8.1 Descripció Detallada	236
6.9 Referència del Fitxer CardCharge.java	236
6.9.1 Descripció Detallada	237
6.10 Referència del Fitxer CardFine.java	237
6.10.1 Descripció Detallada	237
6.11 Referència del Fitxer CardGet.java	237
6.11.1 Descripció Detallada	237
6.12 Referència del Fitxer CardGive.java	237
6.12.1 Descripció Detallada	237
6.13 Referència del Fitxer CardGo.java	238
6.13.1 Descripció Detallada	238
6.14 Referència del Fitxer CardPay.java	238
6.14.1 Descripció Detallada	238
6.15 Referència del Fitxer CPUPlayer.java	238
6.15.1 Descripció Detallada	238
6.16 Referència del Fitxer JocMonopoly.java	238
6.16.1 Descripció Detallada	239
6.17 Referència del Fitxer JSONManager.java	239
6.18 Referència del Fitxer Monopoly.java	239
6.18.1 Descripció Detallada	239
6.19 Referència del Fitxer Movement.java	239
6.19.1 Descripció Detallada	239
6.20 Referència del Fitxer OpActBuy.java	239
6.21 Referència del Fitxer OpActLoan.java	240
6.22 Referència del Fitxer OpActLuckCard.java	240
6.23 Referència del Fitxer OpActSell.java	240
6.24 Referència del Fitxer optionalActions.java	240
6.24.1 Descripció Detallada	240
6.25 Referència del Fitxer OutputManager.java	240
6.25.1 Descripció Detallada	240
6.26 Referència del Fitxer Player.java	241

6.26.1 Descripció Detallada	241
6.27 Referència del Fitxer PlayerLoan.java	241
6.27.1 Descripció Detallada	241
6.28 Referència del Fitxer README.md	241
6.29 Referència del Fitxer TerminalPlayer.java	241
6.29.1 Descripció Detallada	241
6.30 Referència del Fitxer UIFileSelector.java	242
6.30.1 Descripció Detallada	242
Índex alfabètic	243

Capítol 1

Projecte de Programació (GEINF/GEB - UdG)

1.1 Primavera 2021

Aquest directori conté el codi font (fitxers .java)

Capítol 2

Índex Jeràrquic

2.1 Jerarquia de Classes

Aquesta llista d'herència està ordenada toscament, però no completa, de forma alfabètica:

Board	9
Box	27
BoxBet	33
BoxDirectCommand	35
BoxEmpty	38
BoxField	40
BoxStart	56
Card	58
CardCharge	62
CardFine	66
CardGet	71
CardGive	76
CardGo	81
CardPay	85
JFrame	
UIFileSelector	224
JocMonopoly	119
JSONManager	122
Monopoly	128
Movement	144
optionalActions	189
OpActBuy	162
OpActLoan	171
OpActLuckCard	179
OpActSell	182
OutputManager	191
Player	193
CPUPlayer	89
TerminalPlayer	222
PlayerLoan	215

Capítol 3

Índex de Classes

3.1 Llista de Classes

Aquestes són les classes, estructures, unions i interfícies acompanyades amb breus descripcions:

Board	Classe del taulell. Gestiona la informació del taulell, les accions de moviment dels jugadors i altres com la gestió de la fallida	9
Box	Classe de Casella. Gestiona les dades i accions comunes entre tots els diferents tipus de casella	27
BoxBet		33
BoxDirectCommand		35
BoxEmpty		38
BoxField		40
BoxStart		56
Card	Implementa la funcions base i comunes totes les cartes	58
CardCharge	Implementa la funcions de la carta de tipus cobrar	62
CardFine	Implementa la funcions de la carta de tipus multa	66
CardGet	Implementa la funcions de la carta de tipus rebre	71
CardGive	Implementa la funcions de la carta de tipus donar	76
CardGo	Targeta que mou al jugador a una casella determinada en el tauler de Monopoly	81
CardPay	Targeta que fa pagar la quantitat assignada a un jugador	85
CPUPlayer	Jugador controlat per el ordinador	89
JocMonopoly	Main principal del joc del Monopoly	119
JSONManager	JSONManager administra totes les funcions relacionades amb llegir fitxers JSON i també la creació del fitxer de desenvolupament de la partida	122
Monopoly	Classe que s'encarrega d'administrar l'inicialització, els torns i funcionament general i la finalització del joc Monopoly	128

Movement	
Classe que s'encarrega de gestionar les accions que es poden fer en un moviment. Aquest moviment sera fet sobre un Jugador	144
OpActBuy	162
OpActLoan	171
OpActLuckCard	179
OpActSell	182
optionalActions	
Interfície general d'accions opcionals del joc de Monopoly	189
OutputManager	
Classe que s'encarrega de gestionar les sortides en el nostre cas al fitxer de desenvolupament de partida	191
Player	
Classe de Jugador del Monopoly . En aquesta classe hi haura tota la informació del jugador i algunes accions que pot fer el jugador en si	193
PlayerLoan	215
TerminalPlayer	
Jugador controlat per el usuari	222
UIFileSelector	
Implementació d'un selector de fitxers gràfic per poder jugar al Monopoly	224

Capítol 4

Índex de Fitxers

4.1 Llista dels Fitxers

Aquesta és la llista de tots els fitxers acompanyats amb breus descripcions:

Board.java	235
Box.java	235
BoxBet.java	235
BoxDirectCommand.java	236
BoxEmpty.java	236
BoxField.java	236
BoxStart.java	236
Card.java	236
CardCharge.java	236
CardFine.java	237
CardGet.java	237
CardGive.java	237
CardGo.java	238
CardPay.java	238
CPUPlayer.java	238
JocMonopoly.java	238
JSONManager.java	239
Monopoly.java	239
Movement.java	239
OpActBuy.java	239
OpActLoan.java	240
OpActLuckCard.java	240
OpActSell.java	240
optionalActions.java	240
OutputManager.java	240
Player.java	241
PlayerLoan.java	241
TerminalPlayer.java	241
UIFileSelector.java	242

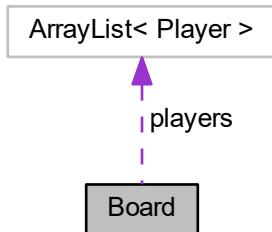
Capítol 5

Documentació de les Classes

5.1 Referència de la Classe Board

Classe del taulell. Gestiona la informació del taulell, les accions de moviment dels jugadors i altres com la gestió de la fallida.

Diagrama de col·laboració per a Board:



Mètodes públics

- **Board ()**
Constructor de Board per defecte.
- void **setBoxesNr (int number)**
Setter del nombre de caselles.
- void **addPlayer (Player player)**
Afegir un jugador nou a la partida.
- void **movePlayer (Player player, int position, ArrayList< String > rewards, OutputManager output)**
Moure el jugador de casella i gestionar el pas per la casella de inici.
- void **addBox (Box box)**
Afageix una casella al taulell.
- boolean **haveOwner (BoxField box)**

Comprova si la casella te propietari.

- **Box getBox (Player player)**

Getter de la casella on estigui el jugador.

- boolean **isBankrupt (Player current_player, int pay_amount, Movement aux)**

Gestiona les accions per quan el jugador està sense diners (en fallida)

- int **numberOfAgrupationField (String group_name)**

Retorna el contador de propietats amb el grup igual a group_name.

- void **transferProperties (Player player_give, Player player_get, Movement movement)**

Transfereix totes les propietats, diners i targetes sort a la banca o a un altre jugador.

- int **getSize ()**

Calcula la mida del taulell.

- String **toString ()**

toString per mostrar l'informació del Taulell per text.

Mètodes Privats

- BoxField **randomField ()**

Genera una casella de terreny de forma aleatoria.

- boolean **haveAvailableFields ()**

Comprova si queden terrenys sense propietari.

- boolean **tryToRunCardToAlive (Player current_player, int missing_money, Scanner scan)**

Intentara executar una targeta de la ma per tal de no declarar-se en fallida.

- boolean **tryToSellToAlive (Player current_player, Movement aux)**

Realitzarà una venda per tal d'afrontar el pagament i no quedar-se en fallida.

- int **selectOption (Player current_player, int pay_amount, int missing_money, Scanner scan)**

Sel·lecciona una opció al pel metode de isBankrupt.

Atributs Privats

- int **boxes_nr**

numero de caselles dels taulell

- SortedMap< Integer, Box > **board** = new TreeMap<>()

Mapa amb les caselles del taulell ordenades.

- ArrayList< Player > **players** = new ArrayList<>()

Llista de jugadors de la partida.

5.1.1 Descripció Detallada

Classe del taulell. Gestiona la informació del taulell, les accions de moviment dels jugadors i altres com la gestió de la fallida.

Definició a la línia 9 del fitxer Board.java.

5.1.2 Documentació del Constructor i el Destructor

5.1.2.1 Board()

Board.Board ()

Constructor de [Board](#) per defecte.

Precondició

true

Postcondició

[Board](#) creat.

Definició a la línia 19 del fitxer Board.java.

5.1.3 Documentació de les Funcions Membre

5.1.3.1 addBox()

```
void Board.addBox (  
    Box box )
```

Afageix una casella al taulell.

Precondició

true

Postcondició

La casella ha estat afegida

Paràmetres

<i>box</i>	Casella nova
------------	--------------

Definició a la línia 115 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.2 addPlayer()

```
void Board.addPlayer (
    Player player )
```

Afegir un jugador nou a la partida.

Precondició

true

Postcondició

El jugador nou ha estat afegit.

Paràmetres

<i>player</i>	Jugador nou.
---------------	--------------

Definició a la línia 37 del fitxer Board.java.

Gràfic de crides a aquesta funció:



5.1.3.3 `getBox()`

```
Box Board.getBox (  
    Player player )
```

Getter de la casella on estigui el jugador.

Precondició

true

Postcondició

La casella on està el jugador ha estat retornada

Paràmetres

<code>player</code>	Jugador actiu
---------------------	---------------

Retorna

La casella de la posició on està el jugador actiu

Definició a la línia 138 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.4 getSize()

```
int Board.getSize ( )
```

Calcula la mida del taulell.

Precondició

true

Postcondició

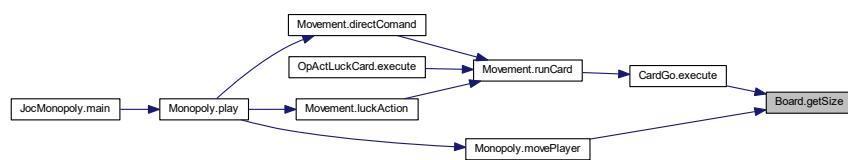
s'ha retornat la mida del taulell

Retorna

la mida del taulell

Definició a la línia 389 del fitxer Board.java.

Gràfic de crides a aquesta funció:



5.1.3.5 haveAvailableFields()

```
boolean Board.haveAvailableFields ( ) [private]
```

Comprova si queden terrenys sense propietari.

Precondició

true

Postcondició

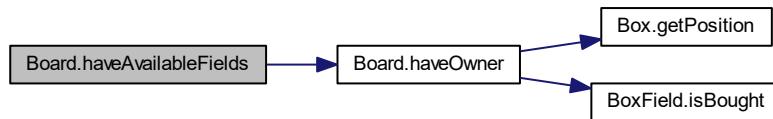
s'ha retornat si queden terrenys sense propietari

Retorna

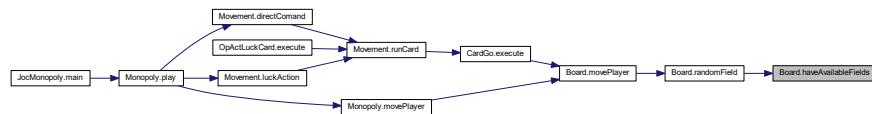
true si queden terrenys sense propietari, false altrament

Definició a la línia 180 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.6 haveOwner()

```
boolean Board.haveOwner (
    BoxField box )
```

Comprova si la casella te propietari.

Precondició

true

Postcondició

s'ha retornat si la casella te propietari

Paràmetres

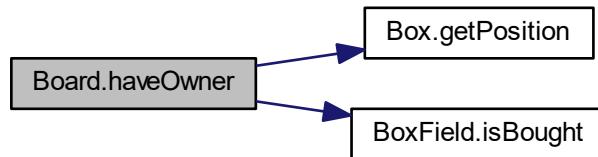
<code>box</code>	Casella de tipus terreny
------------------	--------------------------

Retorna

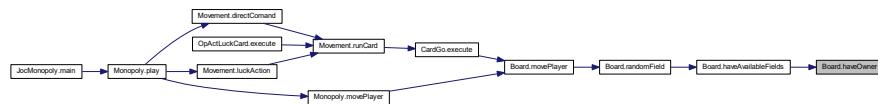
true si la casella te propietari

Definició a la línia 126 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.7 isBankrupt()

```

boolean Board.isBankrupt (
    Player current_player,
    int pay_amount,
    Movement aux )
  
```

Gestiona les accions per quan el jugador està sense diners (en fallida)

Precondició

true

Postcondició

el jugador ha realitzat les accions disponibles i s'ha comprovat si després pot seguir jugant

Retorna

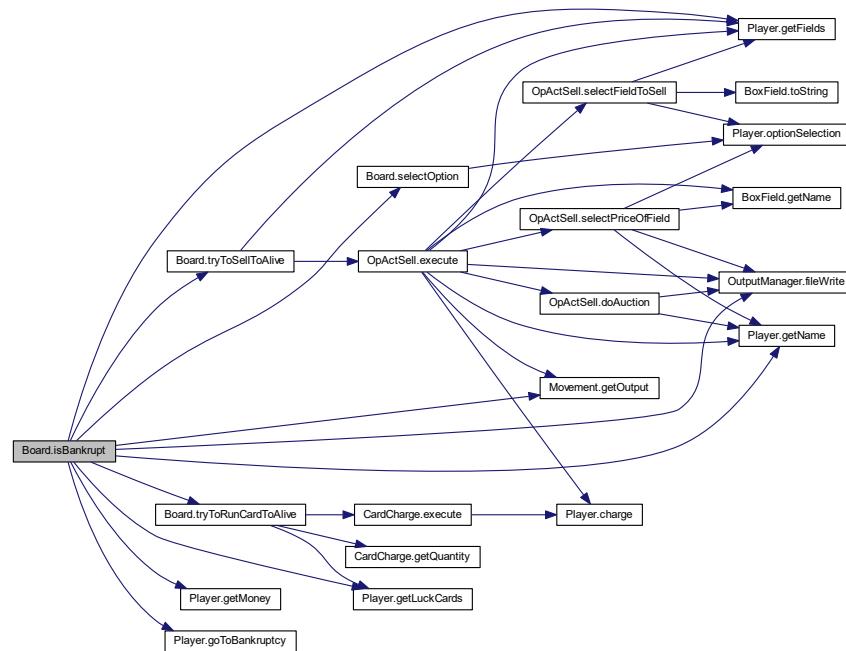
true si el jugador no pot seguir jugant, false altrament

Paràmetres

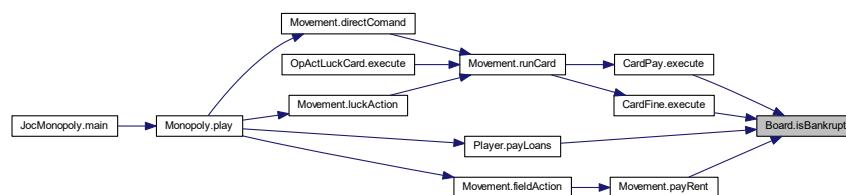
<code>current_player</code>	jugador actual
<code>pay_amount</code>	quantitat a pagar
<code>aux</code>	Classe movement per poder cridar a accions opcionals

Definició a la línia 204 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.8 movePlayer()

```
void Board.movePlayer (
    Player player,
```

```

    int position,
    ArrayList< String > rewards,
    OutputManager output )

```

Moure el jugador de casella i gestionar el pas per la casella de inici.

Precondició

true

Postcondició

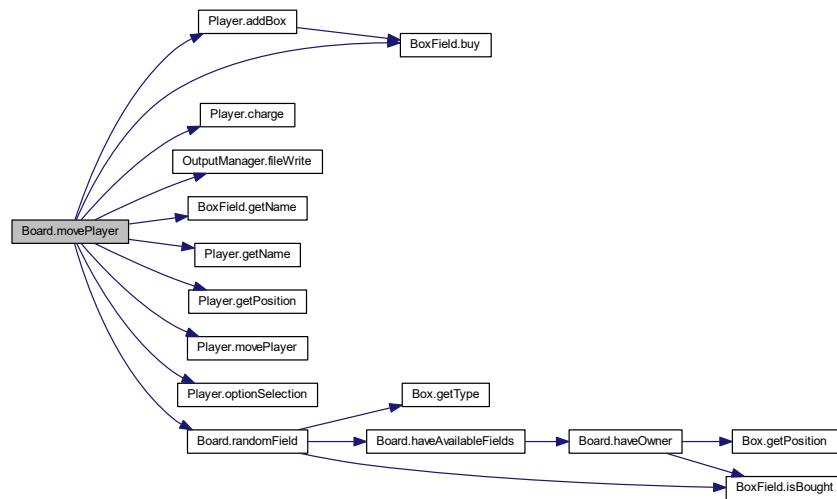
El jugador s'ha mogut a la posició entrada i si ha passat per la casella de sortida, se l'hi ha pagat la recompensa

Paràmetres

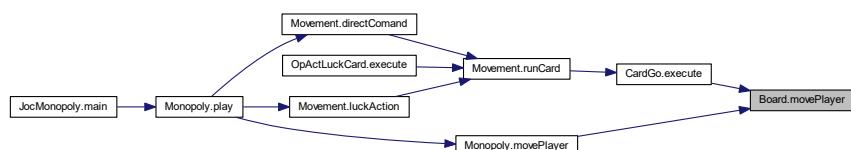
<i>player</i>	Jugador actiu a moure
<i>position</i>	posicio final del jugador
<i>rewards</i>	possibles recompenses de la casella de sortida

Definició a la línia 49 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.9 `numberOfAgrupationField()`

```
int Board.numberOfAgrupationField (
    String group_name )
```

Retorna el contador de propietats amb el grup igual a `group_name`.

Precondició

`true`

Postcondició

El numero de propietats amb `group_name` ha estat retornat.

Paràmetres

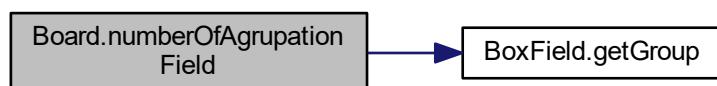
<code>group_name</code>	Grup el qual pertany el Field.
-------------------------	--------------------------------

Retorna

numero de propietats amb el mateix `group_name`.

Definició a la línia 339 del fitxer `Board.java`.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.10 randomField()

```
BoxField Board.randomField ( ) [private]
```

Genera una casella de terreny de forma aleatoria.

Precondició

true

Postcondició

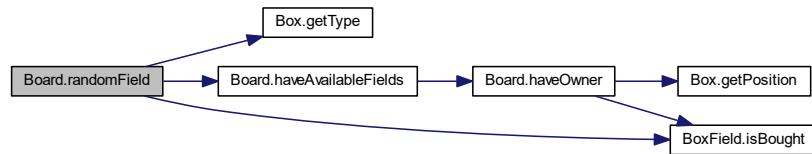
La casella de terreny s'ha retornat

Retorna

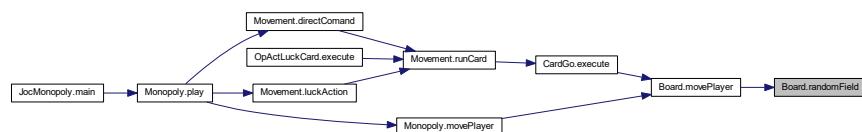
una cassella de terreny aleatoria

Definició a la línia 148 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.11 selectOption()

```
int Board.selectOption (
    Player current_player,
    int pay_amount,
    int missing_money,
    Scanner scan ) [private]
```

Sel·lecciona una opció al pel metode de isBankrupt.

Precondició

true

Postcondició

Retorna la opció sel·leccionada.

Paràmetres

<i>current_player</i>	Jugador actual.
<i>pay_amount</i>	Quantitat que s'ha de pagar.
<i>missing_money</i>	Quantitat que es necessita per afrontar el pagament.

Retorna

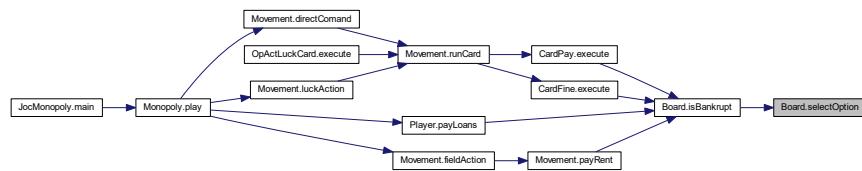
numero de opció triada.

Definició a la línia 317 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.1.3.12 setBoxesNr()**

```
void Board.setBoxesNr (
    int number )
```

Setter del nombre de caselles.

Precondició

true

Postcondició

s'ha guardat el numero de caselles.

Paràmetres

`number` | numero de la casella.

Definició a la línia 27 del fitxer Board.java.

Gràfic de crides a aquesta funció:



5.1.3.13 `toString()`

`String Board.toString ()`

`toString` per mostrar l'informació del Taulell per text.

Precondició

`true`

Postcondició

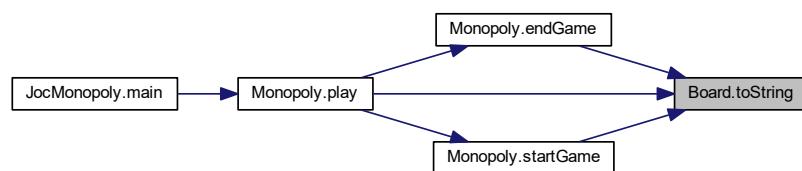
el taulell ha estat mostrat per pantalla amb tota la seva informació per poder seguir la partida adequadament.

Retorna

`salt de linea`

Definició a la línia 400 del fitxer Board.java.

Gràfic de crides a aquesta funció:



5.1.3.14 transferProperties()

```
void Board.transferProperties (
    Player player_give,
    Player player_get,
    Movement movement )
```

Transfereix totes les propietats, diners i targetes sort a la banca o a un altre jugador.

Precondició

Si player_get == null serà la banca, un jugador contrari altrament

Postcondició

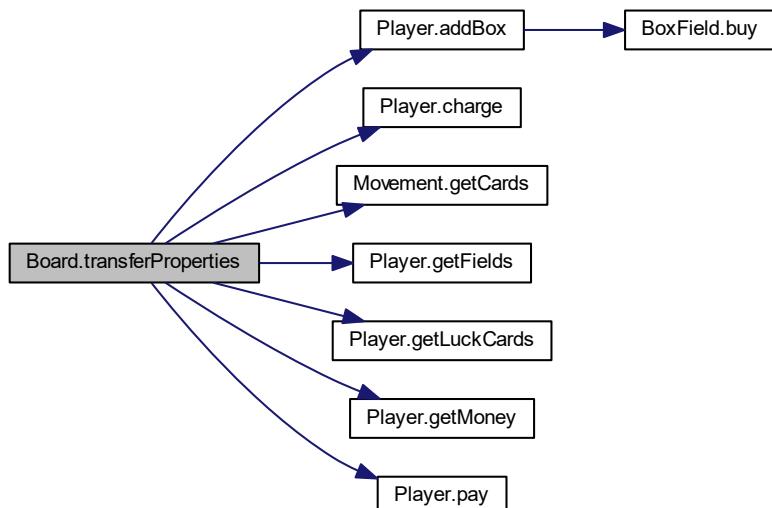
Totes les propietats, diners i targetes, s'han transferit a Jugador o a la banca

Paràmetres

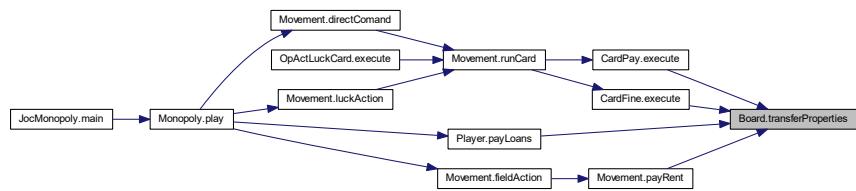
<i>player_give</i>	Jugador que ha de donar les propietats.
<i>player_get</i>	Jugador que ha de rebre les propietatd.
<i>movement</i>	Moviment el qual estava fent.

Definició a la línia 358 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.15 tryToRunCardToAlive()

```

boolean Board.tryToRunCardToAlive (
    Player current_player,
    int missing_money,
    Scanner scan )  [private]
  
```

Intentara executar una targeta de la ma per tal de no declarar-se en fallida.

Precondició

true

Postcondició

Retorna true si s'ha executat una targeta, false altrament.

Paràmetres

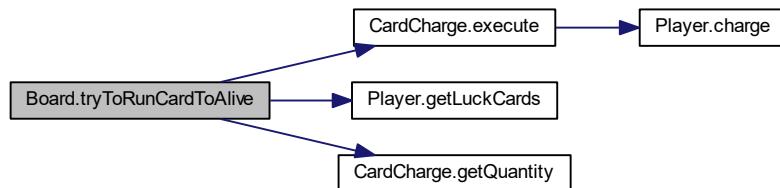
<i>current_player</i>	Jugador actual.
<i>missing_money</i>	Diners que falten per afrontar el pagament.
<i>scan</i>	Scanner d'entrada per l'usuari.

Retorna

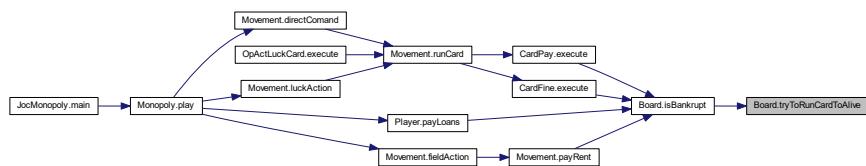
`true` si s'ha pogut executar la carta, `false` altrament.

Definició a la línia 250 del fitxer `Board.java`.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.3.16 tryToSellToAlive()

```
boolean Board.tryToSellToAlive (
    Player current_player,
    Movement aux ) [private]
```

Realitzarà una venda per tal d'afrontar el pagament i no quedar-se en fallida.

Precondició

`true`

Postcondició

Retorna `true` si s'ha pogut vendre.

Paràmetres

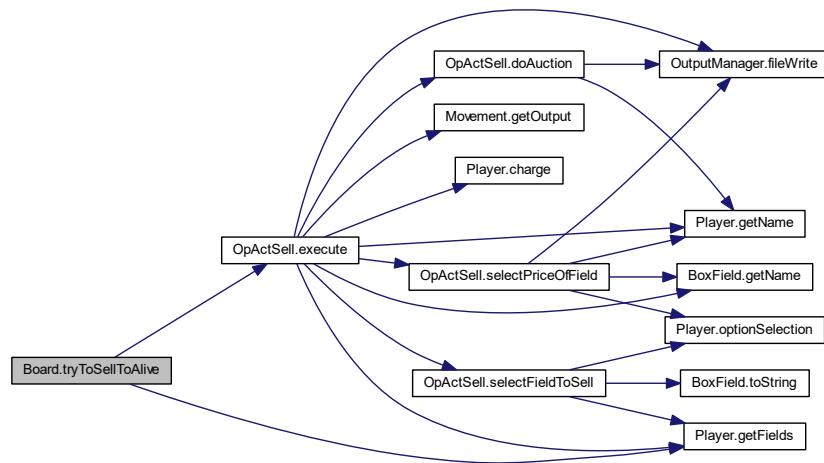
<code>current_player</code>	Jugador actual.
<code>aux</code>	Moviment el qual pertanyen els jugadors.

Retorna

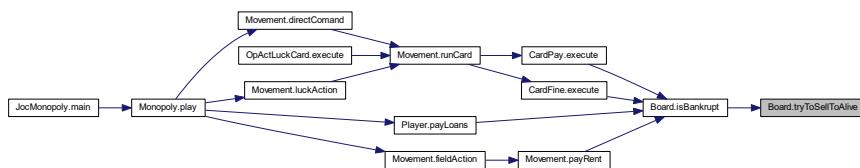
`true` si s'ha pogut realitzar la venda, `false` altrament.

Definició a la línia 296 del fitxer Board.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.1.4 Documentació de les Dades Membre

5.1.4.1 board

```
SortedMap<Integer, Box> Board.board = new TreeMap<>() [private]
```

Mapa amb les caselles del taulell ordenades.

Definició a la línia 11 del fitxer Board.java.

5.1.4.2 boxes_nr

```
int Board.boxes_nr [private]
```

numero de caselles dels taulell

Definició a la línia 10 del fitxer Board.java.

5.1.4.3 players

```
ArrayList<Player> Board.players = new ArrayList<>() [private]
```

Llista de jugadors de la partida.

Definició a la línia 12 del fitxer Board.java.

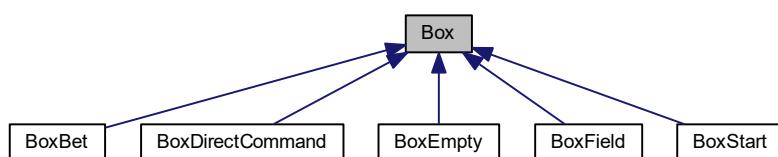
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [Board.java](#)

5.2 Referència de la Classe Box

Classe de Casella. Gestiona les dades i accions comunes entre tots els diferents tipus de casella.

Diagrama d'Herència per a Box:



Mètodes públics

- `Box (int position, String type, String name)`
Constructor de Box.
- `int getPosition ()`
Getter de la posició de la casella.
- `String getType ()`
Getter del tipus de casella.
- `String print (ArrayList< Player > players)`
Sortida per pantalla de la informació basica de la casella i, si n'hi ha, el jugador que està en aquesta.

Atributs Pùblics

- int **position**
Posicio de la casella en el taulell.
- String **type**
Tipus de casella. (Start, Field, Bet, Luck, DirectComand, Empty)

Mètodes Privats

- String **playerInPosition** (ArrayList< **Player** > players)
Busca tots els jugadors que estan en aquesta casella i crea un String amb ells.

Atributs Privats

- final String **name**
Nom de la casella.

5.2.1 Descripció Detallada

Classe de Casella. Gestiona les dades i accions comunes entre tots els diferents tipus de casella.

Definició a la línia 9 del fitxer Box.java.

5.2.2 Documentació del Constructor i el Destructor

5.2.2.1 Box()

```
Box.Box (  
    int position,  
    String type,  
    String name )
```

Constructor de **Box**.

Precondició

true

Postcondició

la casella ha estat creada

Paràmetres

<i>position</i>	Posicio de la casella
<i>type</i>	Tipus de casella. (Start, Field, Bet, Luck, DirectComand, Empty)
<i>name</i>	Nom de la casella

Definició a la línia 23 del fitxer Box.java.

5.2.3 Documentació de les Funcions Membre

5.2.3.1 getPosition()

```
int Box.getPosition ( )
```

Getter de la posicio de la casella.

Precondició

true

Postcondició

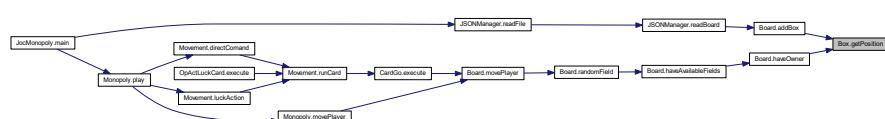
La posicio de la casella ha estat retornada

Retorna

posicio de la casella en el taulell

Definició a la línia 35 del fitxer Box.java.

Gràfic de crides a aquesta funció:



5.2.3.2 getType()

```
String Box.getType ( )
```

Getter del tipus de casella.

Precondició

true

Postcondició

el tipus de casella ha estat retornat

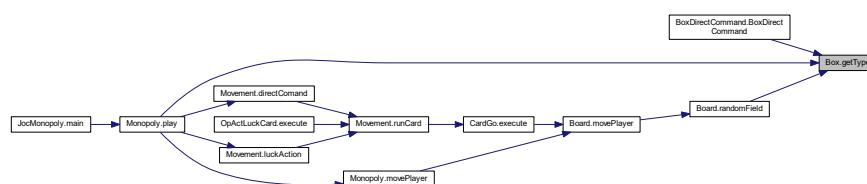
Retorna

tipus de casella

Reimplementat a [BoxStart](#).

Definició a la línia 45 del fitxer Box.java.

Gràfic de crides a aquesta funció:



5.2.3.3 playerInPosition()

```
String Box.playerInPosition (
    ArrayList< Player > players ) [private]
```

Busca tots els jugadors que estan en aquesta casella i crea un String amb ells.

Precondició

true

Postcondició

El string amb el nom de tots els jugadors ha estat creat

Paràmetres

<i>players</i>	llista de jugadors de la partida
----------------	----------------------------------

Retorna

nom dels jugadors que estan en la casella

Definició a la línia 73 del fitxer Box.java.

Gràfic de crides a aquesta funció:

**5.2.3.4 print()**

```
String Box.print (   
    ArrayList< Player > players )
```

Sortida per pantalla de la informació basica de la casella i, si n'hi ha, el jugador que està en aquesta.

Precondició

true

Postcondició

La casella s'ha tret per pantalla

Paràmetres

<i>players</i>	llista de jugadors de la partida
----------------	----------------------------------

Definició a la línia 53 del fitxer Box.java.

Gràfic de crides d'aquesta funció:



5.2.4 Documentació de les Dades Membre

5.2.4.1 name

```
final String Box.name [private]
```

Nom de la casella.

Definició a la línia 13 del fitxer Box.java.

5.2.4.2 position

```
int Box.position
```

Posicio de la casella en el taulell.

Definició a la línia 11 del fitxer Box.java.

5.2.4.3 type

```
String Box.type
```

Tipus de casella. (Start, Field, Bet, Luck, DirectComand, Empty)

Definició a la línia 12 del fitxer Box.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [Box.java](#)

5.3 Referència de la Classe BoxBet

Diagrama d'Herència per a BoxBet:

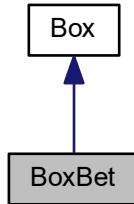
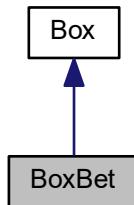


Diagrama de col·laboració per a BoxBet:



Mètodes públics

- `BoxBet (int position, String type)`
Constructor de Bet.
- `int betResult (int quantity, int bet, int dice_result)`
Calcula el resultat de una aposta.

Atributs Privats

- `final double[] combinations = new double[]{0,0,0,1,3,6,10,15,21,26,30,33,35}`
Conté les combinacions acumulades sobre un total de 12 valors (2 daus)

Additional Inherited Members

5.3.1 Descripció Detallada

Definició a la línia 7 del fitxer BoxBet.java.

5.3.2 Documentació del Constructor i el Destructor

5.3.2.1 BoxBet()

```
BoxBet.BoxBet (
    int position,
    String type )
```

Constructor de Bet.

Precondició

`type == "Bet"`

Postcondició

Crea una [Box](#) de tipus aposta

Paràmetres

<code>position</code>	posició de la casella en el taulell
<code>type</code>	tipus de casella == "Bet"

Definició a la línia 18 del fitxer BoxBet.java.

5.3.3 Documentació de les Funcions Membre

5.3.3.1 betResult()

```
int BoxBet.betResult (
    int quanity,
    int bet,
    int dice_result )
```

Calcula el resultat de una aposta.

Precondició

`quantity > 0 i bet > 3`

Postcondició

Retorna el valor final de la aposta si es guanyadora, o -1 si es perdedora

Paràmetres

<i>quantity</i>	quantitat apostada per el jugador
<i>bet</i>	aposta del jugador
<i>dice_result</i>	resultat dels daus obtinguts

Definició a la línia 30 del fitxer BoxBet.java.

Gràfic de crides a aquesta funció:



5.3.4 Documentació de les Dades Membre

5.3.4.1 combinations

```
final double [ ] BoxBet.combinations = new double[ ]{0,0,0,1,3,6,10,15,21,26,30,33,35} [private]
```

Conté les combinacions acomulades sobre un total de 12 valors (2 daus)

Definició a la línia 9 del fitxer BoxBet.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [BoxBet.java](#)

5.4 Referència de la Classe BoxDirectCommand

Diagrama d'Herència per a BoxDirectCommand:

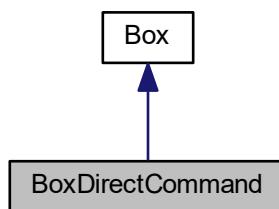
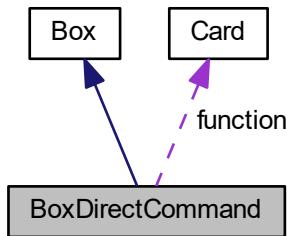


Diagrama de col·laboració per a BoxDirectCommand:



Mètodes públics

- `BoxDirectCommand (int position, String type, Card function)`
Constructor de Buy.
- `Card getCard ()`
Getter de funcio equivalent que te una targeta.

Atributs Privats

- `final Card function`
Targeta a la que equival aquesta casella de directCommand.

Additional Inherited Members

5.4.1 Descripció Detallada

Definició a la línia 8 del fitxer BoxDirectCommand.java.

5.4.2 Documentació del Constructor i el Destructor

5.4.2.1 BoxDirectCommand()

```
BoxDirectCommand.BoxDirectCommand (
    int position,
    String type,
    Card function )
```

Constructor de Buy.

Precondició

```
position > 0 i function != null
```

Postcondició

Crea una casella de comanda directe amb la funció function.

Definició a la línia 17 del fitxer BoxDirectCommand.java.

Gràfic de crides d'aquesta funció:



5.4.3 Documentació de les Funcions Membre

5.4.3.1 getCard()

```
Card BoxDirectCommand.getCard ()
```

Getter de funció equivalent que te una targeta.

Precondició

```
true
```

Postcondició

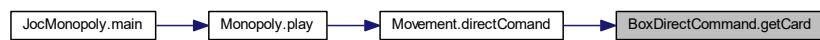
La targeta ha sigut retornada.

Retorna

```
targeta amb una funció function determinada.
```

Definició a la línia 28 del fitxer BoxDirectCommand.java.

Gràfic de crides a aquesta funció:



5.4.4 Documentació de les Dades Membre

5.4.4.1 function

```
final Card BoxDirectCommand.function [private]
```

Targeta a la que equival aquesta casella de directCommand.

Definició a la línia 10 del fitxer BoxDirectCommand.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [BoxDirectCommand.java](#)

5.5 Referència de la Classe BoxEmpty

Diagrama d'Herència per a BoxEmpty:

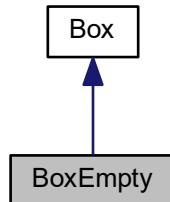
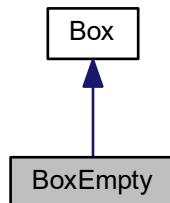


Diagrama de col·laboració per a BoxEmpty:



Mètodes públics

- [BoxEmpty \(int position, String type\)](#)

Constructor de Empty.

Additional Inherited Members

5.5.1 Descripció Detallada

Definició a la línia 8 del fitxer BoxEmpty.java.

5.5.2 Documentació del Constructor i el Destructor

5.5.2.1 BoxEmpty()

```
BoxEmpty.BoxEmpty (
    int position,
    String type )
```

Constructor de Empty.

Precondició

true

Postcondició

Crea una casella Empty amb les propietats d'entrada i heredades.

Paràmetres

<i>position</i>	posició en el tauler de la casella.
<i>type</i>	tipus de casella a la que pertany.

Definició a la línia 17 del fitxer BoxEmpty.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [BoxEmpty.java](#)

5.6 Referència de la Classe BoxField

Diagrama d'Herència per a BoxField:

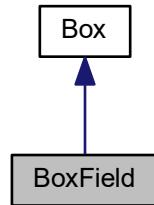
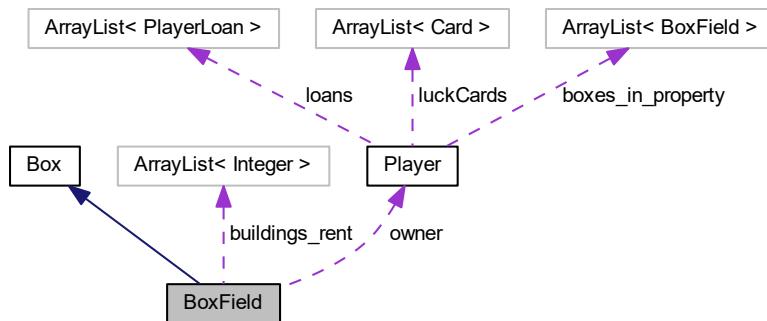


Diagrama de col·laboració per a BoxField:



Mètodes públics

- `BoxField (int position, String type, String name, int price, String group, int basic_rent, int group_rent, String buildable, int max_buildings, int building_price, boolean hotel, int hotel_price, ArrayList< Integer > buildings_rent, int hotel_rent)`

Constructor de terreny Field.
- `void buy (Player owner)`

Assigna un Jugador Player com a propietari owner i actualitza estat del terreny.
- `void sell ()`

Elimina el propietari owner del terreny.
- `int getPrice ()`

Getter del preu del terreny Field.
- `int getRent ()`

Getter del lloguer del terreny Field dependent de les edificacions que tingui.
- `Player getOwner ()`

- **String getOwner ()**
Getter del propietari owner del terreny.
- **String getGroup ()**
Getter de la agrupació del terreny.
- **void build (int quantity)**
Afegeix quantity com a edificis edificats al terreny Field.
- **String houseBuildableType ()**
Consulta si es pot construir cases en el terreny.
- **boolean houseBuildable ()**
Consulta si es pot construir cases en el terreny.
- **boolean hotelBuildable ()**
Consulta si es pot contruir un hotel a terreny.
- **int priceToBuild ()**
Consulta el preu per construir el el edifici que correspon.
- **int numberOfHouseBuildable ()**
Consulta el nombre de cases que es pot construir en un terreny.
- **boolean isBought ()**
Consulta si la propietat esta comprada.
- **String getName ()**
Getter del nom de la terreny.
- **int getNumberOfApartments ()**
Getter de nombre d'edificacions construïdes.
- **int getNumberOfHotels ()**
Consulta el nombre d'hotels que estan construïts.
- **String toString ()**
toString per mostrar la descripció del terreny Field en text.

Atributs Privats

- **final String name**
Nom del terreny.
- **final int price**
Preu del terreny.
- **final String group**
Color d'agrupació.
- **final int basic_rent**
Preu lloguer sense agrupació.
- **final int group_rent**
Preu lloguer agrupació.
- **final String buildable**
Tipus de construcció.
- **int max_buildings**
Nombre màxim d'apartaments construïbles.
- **final int building_price**
Preu de cada apartament.
- **final boolean hotel**
Es pot contruir hotel, true si, false no.
- **final int hotel_price**
Preu de l'hotel.
- **final ArrayList< Integer > buildings_rent**
Lloguer depenent de les construccions.

- final int **hotel_rent**
Lloguer del hotel.
- **Player owner**
Jugador propietari.
- int **builded** = 0
Nom de edificacions construïdes.
- boolean **bought** = false
Estat de comprat o no.

Additional Inherited Members

5.6.1 Descripció Detallada

Definició a la línia 9 del fitxer BoxField.java.

5.6.2 Documentació del Constructor i el Destructor

5.6.2.1 BoxField()

```
BoxField.BoxField (
    int position,
    String type,
    String name,
    int price,
    String group,
    int basic_rent,
    int group_rent,
    String buildable,
    int max_buildings,
    int building_price,
    boolean hotel,
    int hotel_price,
    ArrayList< Integer > buildings_rent,
    int hotel_rent )
```

Constructor de terreny Field.

Precondició

true

Postcondició

Crea un terreny amb tots els atributs entrats.

Definició a la línia 32 del fitxer BoxField.java.

5.6.3 Documentació de les Funcions Membre

5.6.3.1 build()

```
void BoxField.build (
    int quantity )
```

Afegeix `quantity` com a edificis edificats al terreny `Field`.

Precondició

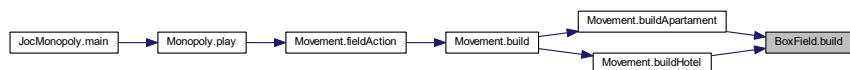
Es necessari saber que el usuari ja ha pagat el preu corresponent al apartament o hotel segons toqui i que possible construir

Postcondició

S'ha construit `quantity` apartaments al terreny.

Definició a la línia 129 del fitxer `BoxField.java`.

Gràfic de crides a aquesta funció:



5.6.3.2 buy()

```
void BoxField.buy (
    Player owner )
```

Assigna un Jugador `Player` com a propietari `owner` i actualitza estat del terreny.

Precondició

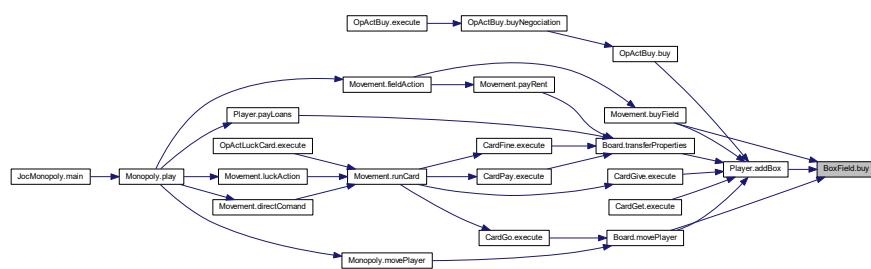
`owner != null`

Postcondició

Propietari assignat i estat bought = true.

Definició a la línia 54 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.3 getGroup()

```
String BoxField.getGroup( )
```

Getter de la agrupació del terreny.

Precondició

true

Postcondició

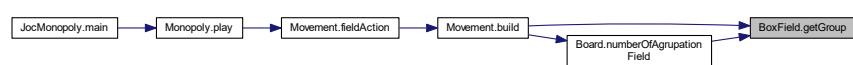
Retorna el el color de la agrupació del terreny

Retorna

agrupació del terreny

Definició a la línia 119 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.4 getName()

```
String BoxField.getName ( )
```

Getter del nom de la terreny.

Precondició

true

Postcondició

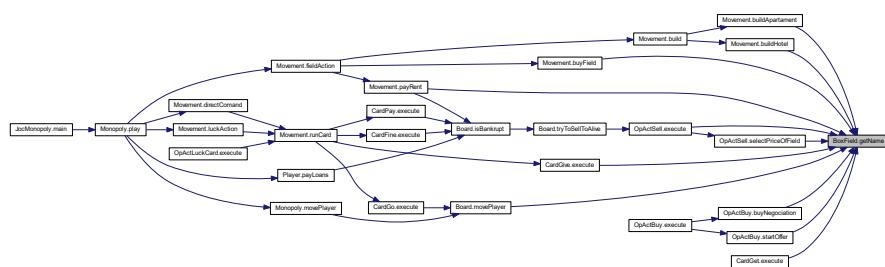
Retorna el nom del terreny.

Retorna

string del nom name del terreny

Definició a la línia 202 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.5 getNumberOfApartments()

```
int BoxField.getNumberOfApartments ( )
```

Getter de nombre d'edificacions construïdes.

Precondició

true

Postcondició

Retorna el nombre d'apartaments construïts.

Retorna

nombre de edificacions a Field.

Definició a la línia 212 del fitxer BoxField.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.6.3.6 getNumberOfHotels()**

```
int BoxField.getNumberOfHotels ( )
```

Consulta el nombre d'hotels que estan construïts.

Precondició

true

Postcondició

Retorna el nombre d'hotels construïts.

Retorna

nombre d'hotels a Field.

Definició a la línia 223 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.7 getOwner()

```
Player BoxField.getOwner ( )
```

Getter del propietari `owner` del terreny.

Precondició

true

Postcondició

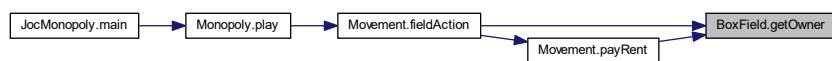
Retorna el propietari `owner` del terreny `Field`.

Retorna

Jugador `owner` propietari de `Field`.

Definició a la línia 109 del fitxer `BoxField.java`.

Gràfic de crides a aquesta funció:



5.6.3.8 getPrice()

```
int BoxField.getPrice ( )
```

Getter del preu del terreny `Field`.

Precondició

true

Postcondició

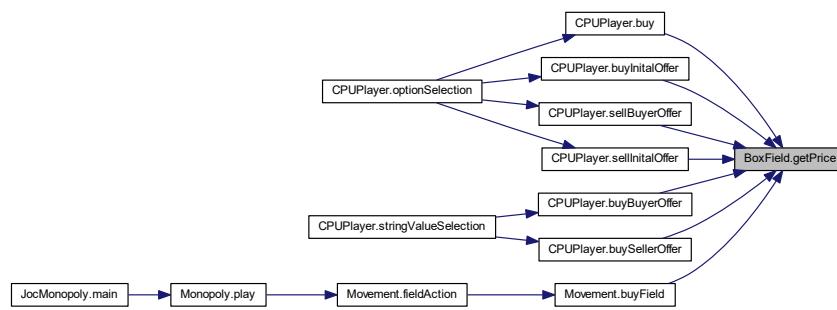
Retorna el preu del terreny `Field`.

Retorna

preu price del terreny.

Definició a la línia 75 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:

**5.6.3.9 getRent()**

```
int BoxField.getRent ()
```

Getter del lloguer del terreny Field depenent de les edificacions que tingui.

Precondició

true

Postcondició

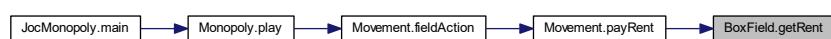
Retorna el lloguer del terreny Field que correspon.

Retorna

lloguer del terreny Field depenent de builded.

Definició a la línia 85 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.10 hotelBuildable()

```
boolean BoxField.hotelBuildable ( )
```

Consulta si es pot construir un hotel a terreny.

Precondició

true

Postcondició

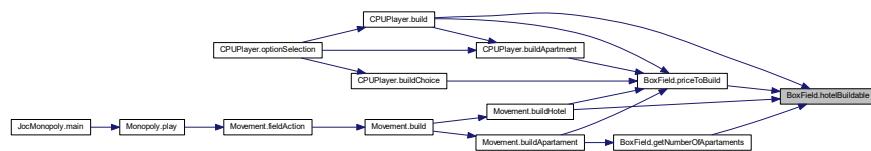
Retorna true si es pot construir un hotel a la propietat, false altrament,

Retorna

true si es pot construir un hotel, false altrament.

Definició a la línia 159 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.11 houseBuildable()

```
boolean BoxField.houseBuildable ( )
```

Consulta si es pot construir cases en el terreny.

Precondició

true

Postcondició

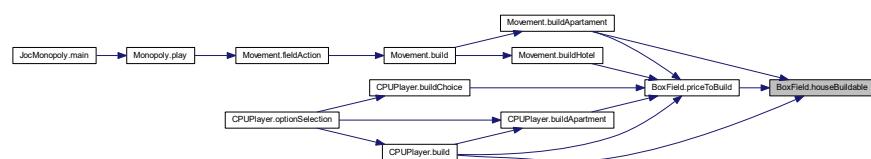
Retorna true si a la propietat s'hi pot construir cases, false altrament.

Retorna

true si es pot construir cases, false altrament

Definició a la línia 149 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.12 houseBuildableType()

```
String BoxField.houseBuildableType ( )
```

Consulta si es pot construir cases en el terreny.

Precondició

true

Postcondició

Retorna true si a la propietat s'hi pot construir cases, false altrament.

Retorna

true si es pot construir cases, false altrament

Definició a la línia 139 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.13 isBought()

```
boolean BoxField.isBought ( )
```

Consulta si la propietat esta comprada.

Precondició

true

Postcondició

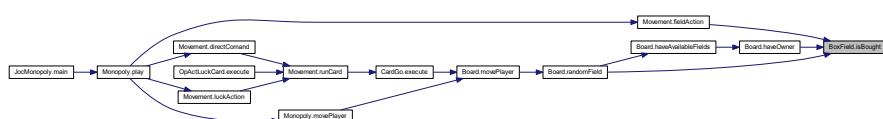
Retorna true si la propietat esta comprada, false altrament.

Retorna

true si bought = true, false altrament.

Definició a la línia 192 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.14 `numberOfHouseBuildable()`

```
int BoxField.numberOfHouseBuildable ( )
```

Consulta el nombre de cases que es pot construir en un terreny.

Precondició

`houseBuildable = true`

Postcondició

Retorna el numero de cases possibles a construir.

Retorna

numero de cases per construir.

Definició a la línia 182 del fitxer BoxField.java.

Gràfic de crides a aquesta funció:



5.6.3.15 `priceToBuild()`

```
int BoxField.priceToBuild ( )
```

Consulta el preu per construir el edifici que correspon.

Precondició

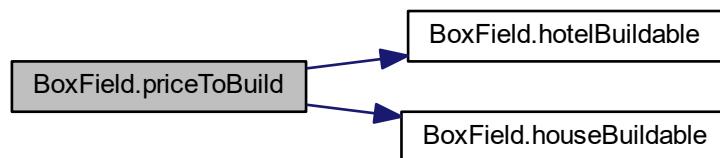
`true`

Postcondició

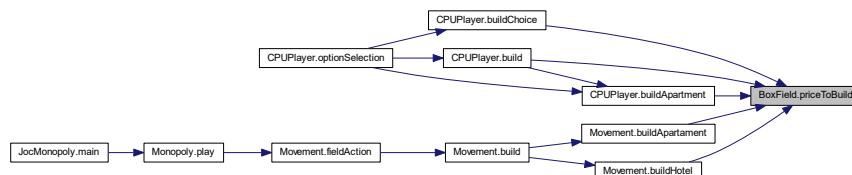
Retorna el preu d'una casa `building_price` si pot ser construïda, `hotel_price` si només pot construir-se un hotel, -1 altrament.

Definició a la línia 170 del fitxer `BoxField.java`.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.6.3.16 sell()**

```
void BoxField.sell( )
```

Elimina el propietari `owner` del terreny.

Precondició

`true`

Postcondició

Elimina le Jugador com a propietari `owner` i canvia el estat `bought` a `false`.

Definició a la línia 64 del fitxer `BoxField.java`.

5.6.3.17 `toString()`

```
String BoxField.toString ( )
```

`toString` per mostrar la descripció del terreny `Field` en text.

Precondició

true

Postcondició

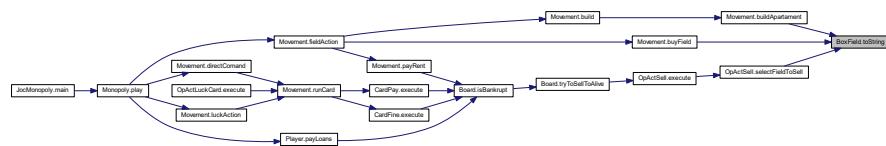
Mostra els detalls de `Field`.

Retorna

salt de linea.

Definició a la línia 235 del fitxer `BoxField.java`.

Gràfic de crides a aquesta funció:



5.6.4 Documentació de les Dades Membre

5.6.4.1 `basic_rent`

```
final int BoxField.basic_rent [private]
```

Preu lloguer sense agrupació.

Definició a la línia 14 del fitxer `BoxField.java`.

5.6.4.2 `bought`

```
boolean BoxField.bought = false [private]
```

Estat de comprat o no.

Definició a la línia 25 del fitxer `BoxField.java`.

5.6.4.3 buildable

```
final String BoxField.buildable [private]
```

Tipus de construcció.

Definició a la línia 16 del fitxer BoxField.java.

5.6.4.4 buildded

```
int BoxField.buildded = 0 [private]
```

Nombre de edificacions construïdes.

Definició a la línia 24 del fitxer BoxField.java.

5.6.4.5 building_price

```
final int BoxField.building_price [private]
```

Preu de cada apartament.

Definició a la línia 18 del fitxer BoxField.java.

5.6.4.6 buildings_rent

```
final ArrayList<Integer> BoxField.buildings_rent [private]
```

Lloguer dependent de les construccions.

Definició a la línia 21 del fitxer BoxField.java.

5.6.4.7 group

```
final String BoxField.group [private]
```

Color d'agrupació.

Definició a la línia 13 del fitxer BoxField.java.

5.6.4.8 group_rent

```
final int BoxField.group_rent [private]
```

Preu lloguer agrupació.

Definició a la línia 15 del fitxer BoxField.java.

5.6.4.9 hotel

```
final boolean BoxField.hotel [private]
```

Es pot construir hotel, true si, false no.

Definició a la línia 19 del fitxer BoxField.java.

5.6.4.10 hotel_price

```
final int BoxField.hotel_price [private]
```

Preu de l'hotel.

Definició a la línia 20 del fitxer BoxField.java.

5.6.4.11 hotel_rent

```
final int BoxField.hotel_rent [private]
```

Lloguer del hotel.

Definició a la línia 22 del fitxer BoxField.java.

5.6.4.12 max_buildings

```
int BoxField.max_buildings [private]
```

Nombrà màxim d' apartaments construïbles.

Definició a la línia 17 del fitxer BoxField.java.

5.6.4.13 name

```
final String BoxField.name [private]
```

Nom del terreny.

Definició a la línia 11 del fitxer BoxField.java.

5.6.4.14 owner

```
Player BoxField.owner [private]
```

Jugador propietari.

Definició a la línia 23 del fitxer BoxField.java.

5.6.4.15 price

```
final int BoxField.price [private]
```

Preu del terreny.

Definició a la línia 12 del fitxer BoxField.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [BoxField.java](#)

5.7 Referència de la Classe BoxStart

Diagrama d'Herència per a BoxStart:

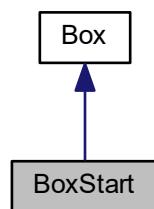
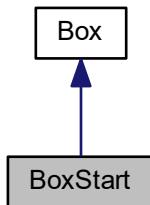


Diagrama de col·laboració per a BoxStart:



Mètodes públics

- **BoxStart** (int *position*, String *type*)
Constructor de Start.
- String **getType** ()
Retorna el tipus \type de casella a la que pertany.

Additional Inherited Members

5.7.1 Descripció Detallada

Definició a la línia 9 del fitxer BoxStart.java.

5.7.2 Documentació del Constructor i el Destructor

5.7.2.1 BoxStart()

```
BoxStart.BoxStart (
    int position,
    String type )
```

Constructor de Start.

Precondició

true

Postcondició

Crea una casella Start

Paràmetres

<i>position</i>	posició de la casella en el tauler.
<i>type</i>	tipus de casella a la que pertany.

Definició a la línia 18 del fitxer BoxStart.java.

5.7.3 Documentació de les Funcions Membre

5.7.3.1 `getType()`

`String BoxStart.getType ()`

Retorna el tipus \type de casella a la que pertany.

Precondició

`true`

Postcondició

El tipus `type` de casella ha estat retornat.

Retorna

`String` amb el `type` de casella.

Reimplementat de [Box](#).

Definició a la línia 28 del fitxer BoxStart.java.

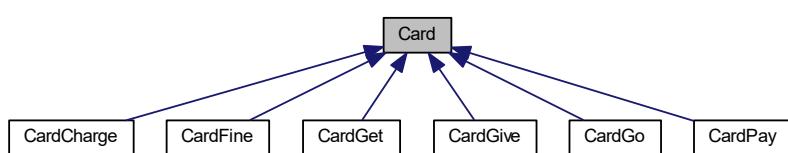
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [BoxStart.java](#)

5.8 Referència de la Classe Card

Implementa la funcions base i comunes totes les cartes.

Diagrama d'Herència per a Card:



Mètodes públics

- `Card` (String `type`, boolean `postposable`)

Constructor de `Card`.

- boolean `isPostposable` ()

Comprova si la carta es posposable.

- String `getType` ()

Retorna el tipus de la carta.

Atributs Privats

- final String `type`
- final boolean `postposable`

true si es posposable, false altrament

5.8.1 Descripció Detallada

Implementa la funcions base i comunes totes les cartes.

Definició a la línia 8 del fitxer Card.java.

5.8.2 Documentació del Constructor i el Destructor

5.8.2.1 Card()

```
Card.Card (
    String type,
    boolean postposable )
```

Constructor de `Card`.

Precondició

`true`

Postcondició

La carta ha estat creada

Paràmetres

<code>type</code>	tipus de carta
<code>postposable</code>	true si es posposable, false altrament

Definició a la línia 19 del fitxer Card.java.

5.8.3 Documentació de les Funcions Membre

5.8.3.1 getType()

```
String Card.getType ( )
```

Retorna el tipus de la carta.

Precondició

true

Postcondició

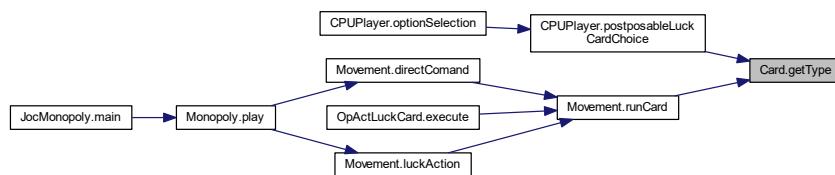
el tipus de la carta s'ha retornat

Retorna

el tipus de la carta

Definició a la línia 40 del fitxer Card.java.

Gràfic de crides a aquesta funció:



5.8.3.2 isPostposable()

```
boolean Card.isPostposable ( )
```

Comprova si la carta es posposable.

Precondició

true

Postcondició

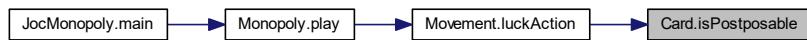
s'ha comprovat si la carta es posposable

Retorna

treu si la carta es posposable, false altrament

Definició a la línia 30 del fitxer Card.java.

Gràfic de crides a aquesta funció:



5.8.4 Documentació de les Dades Membre

5.8.4.1 postposable

```
final boolean Card.postposable [private]
```

true si es posposable, false altrament

Definició a la línia 10 del fitxer Card.java.

5.8.4.2 type

```
final String Card.type [private]
```

Definició a la línia 9 del fitxer Card.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [Card.java](#)

5.9 Referència de la Classe CardCharge

Implementa la funcions de la carta de tipus cobrar.

Diagrama d'Herència per a CardCharge:

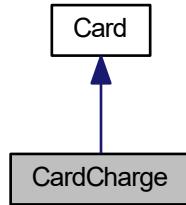
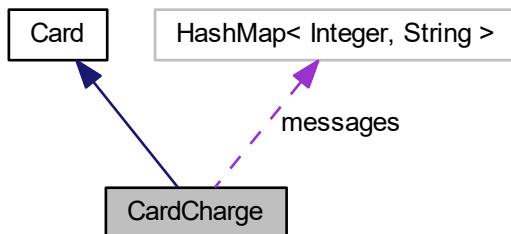


Diagrama de col·laboració per a CardCharge:



Mètodes públics

- **CardCharge** (boolean `postposable`, int `quantity`)
Constructor de CardCharge.
- void **execute** (Player `current_player`)
Execucio de la carta.
- int **getQuantity** ()
Getter de la quantitat a cobrar.
- String **toString** ()
Sortida.

Atributs Privats

- final int **quantity**
quantitat a pagar
- String **message**
Missatge de la carta.
- final HashMap< Integer, String > **messages** = new HashMap<>()
llista de missatges possibles

5.9.1 Descripció Detallada

Implementa la funcions de la carta de tipus cobrar.

Definició a la línia 10 del fitxer CardCharge.java.

5.9.2 Documentació del Constructor i el Destructor

5.9.2.1 CardCharge()

```
CardCharge.CardCharge (
    boolean postposable,
    int quantity )
```

Constructor de [CardCharge](#).

Precondició

`true`

Postcondició

La carta ha estat creada

Paràmetres

<code><i>postposable</i></code>	true si es posposable, false altrament
<code><i>quantity</i></code>	quantitat a pagar

Definició a la línia 22 del fitxer CardCharge.java.

5.9.3 Documentació de les Funcions Membre

5.9.3.1 execute()

```
void CardCharge.execute (
    Player current_player )
```

Execució de la carta.

Precondició

true

Postcondició

El jugador ha rebut la quantitat assignada

Paràmetres

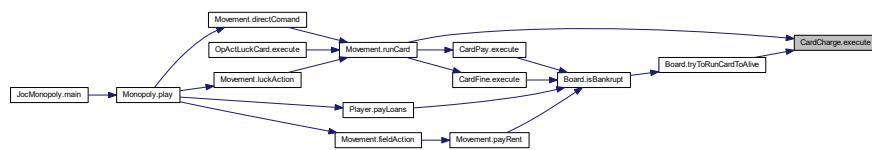
<i>current_player</i>	jugador actiu
-----------------------	---------------

Definició a la línia 47 del fitxer CardCharge.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.9.3.2 getQuantity()

```
int CardCharge.getQuantity ( )
```

Getter de la quantitat a cobrar.

Precondició

true

Postcondició

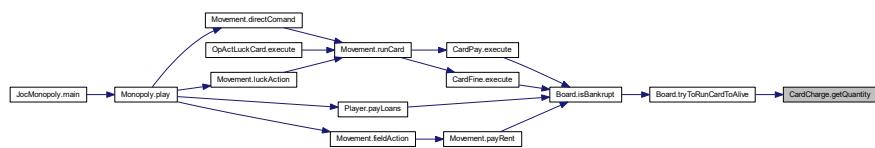
la quantitat a cobrar ha estat retornada

Retorna

la quantitat a cobrar

Definició a la línia 58 del fitxer CardCharge.java.

Gràfic de crides a aquesta funció:



5.9.3.3 `toString()`

```
String CardCharge.toString ( )
```

Sortida.

Precondició

true

Postcondició

el missatge ha estat retornat

Retorna

missatge de sortida

Definició a la línia 66 del fitxer CardCharge.java.

5.9.4 Documentació de les Dades Membre

5.9.4.1 message

```
String CardCharge.message [private]
```

Missatge de la carta.

Definició a la línia 12 del fitxer CardCharge.java.

5.9.4.2 messages

```
final HashMap<Integer, String> CardCharge.messages = new HashMap<>() [private]
```

LLista de missatges possibles

Definició a la línia 13 del fitxer CardCharge.java.

5.9.4.3 quantity

```
final int CardCharge.quantity [private]
```

quantitat a pagar

Definició a la línia 11 del fitxer CardCharge.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [CardCharge.java](#)

5.10 Referència de la Classe CardFine

Implementa la funcions de la carta de tipus multa.

Diagrama d'Herència per a CardFine:

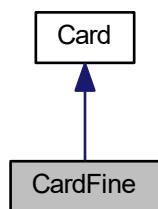
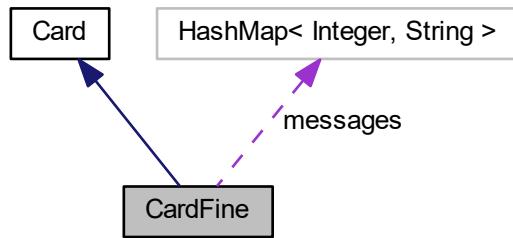


Diagrama de col·laboració per a CardFine:



Mètodes públics

- `CardFine (boolean postposable, int quantity)`
Constructor de CardFine.
- `void execute (Board board, Player current_player, Movement aux)`
Execució de la carta.
- `String toString ()`
Sortida.

Atributs Privats

- `final int quantity`
quantitat a pagar
- `String message`
Missatge de la carta.
- `final HashMap< Integer, String > messages = new HashMap<>()`
llista de missatges possibles

5.10.1 Descripció Detallada

Implementa la funcions de la carta de tipus multa.

Definició a la línia 10 del fitxer CardFine.java.

5.10.2 Documentació del Constructor i el Destructor

5.10.2.1 CardFine()

```
CardFine.CardFine (
    boolean postposable,
    int quantity )
```

Constructor de [CardFine](#).

Precondició

`true`

Postcondició

La carta ha estat creada

Paràmetres

<code><i>postposable</i></code>	true si es posposable, false altrament
<code><i>quantity</i></code>	quantitat a pagar

Definició a la línia 22 del fitxer `CardFine.java`.

5.10.3 Documentació de les Funcions Membre

5.10.3.1 execute()

```
void CardFine.execute (
    Board board,
    Player current_player,
    Movement aux )
```

Execució de la carta.

Precondició

`true`

Postcondició

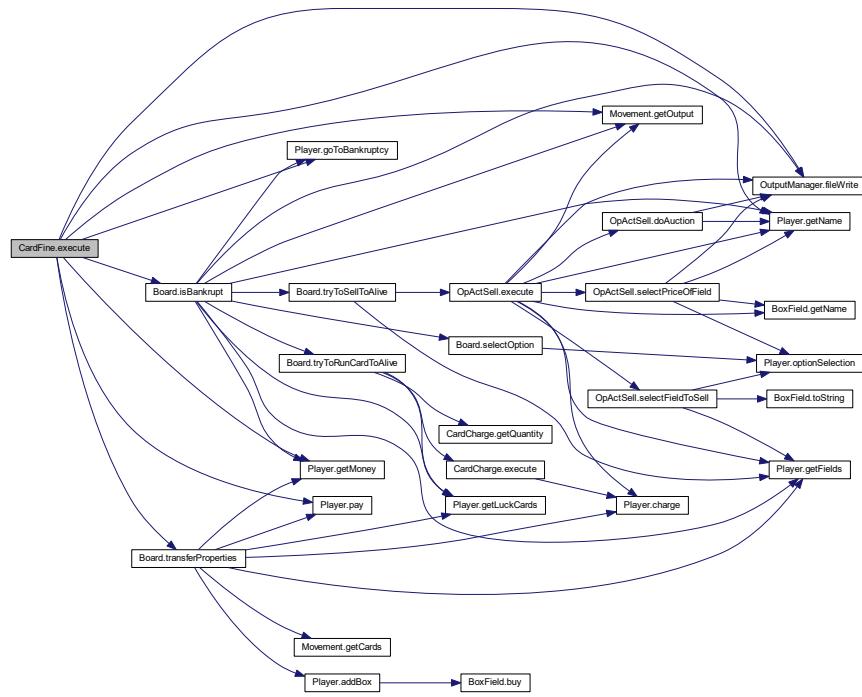
La multa ha estat pagada

Paràmetres

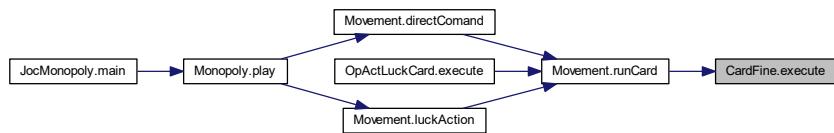
<code><i>board</i></code>	taulell
<code><i>current_player</i></code>	jugador actiu
<code><i>aux</i></code>	Movement que crida Buy, en aquesta implementació, <code>m</code> no és usada però s'ha de passar.

Definició a la línia 49 del fitxer CardFine.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.10.3.2 `toString()`

```
String CardFine.toString ( )
```

Sortida.

Precondició

true

Postcondició

el missatge ha estat retornat

Retorna

missatge de sortida

Definició a la línia 77 del fitxer CardFine.java.

5.10.4 Documentació de les Dades Membre

5.10.4.1 message

```
String CardFine.message [private]
```

Missatge de la carta.

Definició a la línia 12 del fitxer CardFine.java.

5.10.4.2 messages

```
final HashMap<Integer, String> CardFine.messages = new HashMap<>() [private]
```

llista de missatges possibles

Definició a la línia 13 del fitxer CardFine.java.

5.10.4.3 quantity

```
final int CardFine.quantity [private]
```

quantitat a pagar

Definició a la línia 11 del fitxer CardFine.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [CardFine.java](#)

5.11 Referència de la Classe CardGet

Implementa la funcions de la carta de tipus rebre.

Diagrama d'Herència per a CardGet:

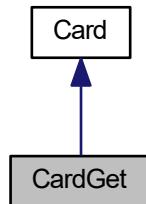
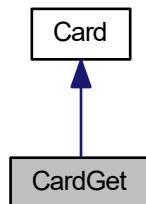


Diagrama de col·laboració per a CardGet:



Mètodes públics

- **CardGet** (boolean postposable)
Constructor de CardGet.
- void **execute** (ArrayList< Player > players, Player current_player)
Execucio de la carta.
- String **toString** ()
Sortida.

Mètodes Privats

- Player **playerSelect** (ArrayList< Player > players, Player current_player)
Mostra tots els jugadors i deixa escollir a un d'ells.
- BoxField **fieldSelect** (Player choosed)
Mostra tots els terreyns del jugador i deixa escollir a un d'ells.

5.11.1 Descripció Detallada

Implementa la funcions de la carta de tipus rebre.

Definició a la línia 9 del fitxer CardGet.java.

5.11.2 Documentació del Constructor i el Destructor

5.11.2.1 CardGet()

```
CardGet.CardGet (  
    boolean postposable )
```

Constructor de [CardGet](#).

Precondició

true

Postcondició

La carta ha estat creada

Paràmetres

<i>postposable</i>	true si es posposable, false altrament
--------------------	--

Definició a la línia 17 del fitxer CardGet.java.

5.11.3 Documentació de les Funcions Membre

5.11.3.1 execute()

```
void CardGet.execute (   
    ArrayList< Player > players,  
    Player current_player )
```

Execució de la carta.

Precondició

true

Postcondició

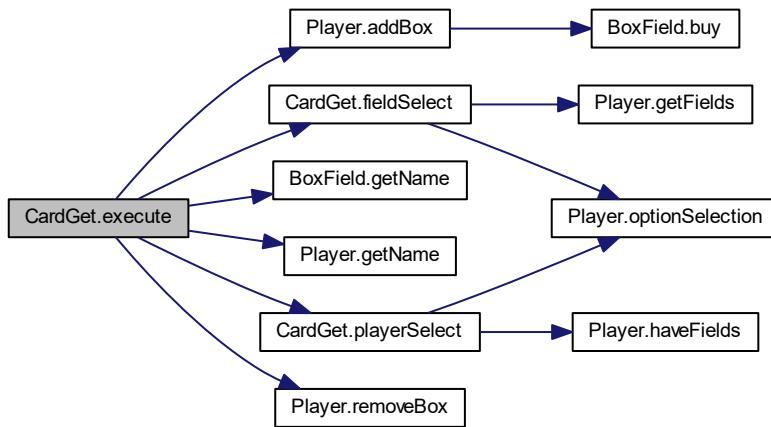
la propietat ha estat rebuda

Paràmetres

<i>players</i>	llista de jugadors de la partida
<i>current_player</i>	jugador actiu

Definició a la línia 28 del fitxer CardGet.java.

Gràfic de crides d'aquesta funció:



5.11.3.2 fieldSelect()

```
BoxField CardGet.fieldSelect (
    Player choosed ) [private]
```

Mostra tots els terreys del jugador i deixa escollir a un d'ells.

Paràmetres

<i>choosed</i>	Jugador que ha de donar el terreny
----------------	------------------------------------

Retorna

el terreny seleccionat per el jugador o null si no es pot

Precondició

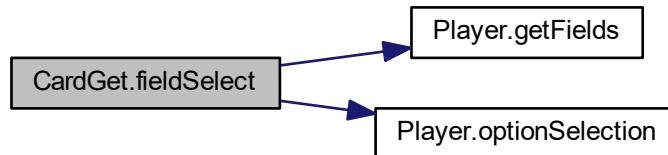
choosed te algun terreny (condicio heredada de playerSelect)

Postcondició

El terreny seleccionat ha estat retornat, o en la seva absència, s'ha retornat un terreny null (no retornarà mai null per la precondició)

Definició a la línia 77 del fitxer CardGet.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.11.3.3 playerSelect()

```

Player CardGet.playerSelect (
    ArrayList< Player > players,
    Player current_player ) [private]
  
```

Mostra tots els jugadors i deixa escollir a un d'ells.

Paràmetres

<code>players</code>	ArrayList de jugadors que estan jugant al Monopoly.
<code>current_player</code>	Jugador que reb el terreny

Retorna

el jugador seleccionat o null si no n'hi ha cap

Precondició

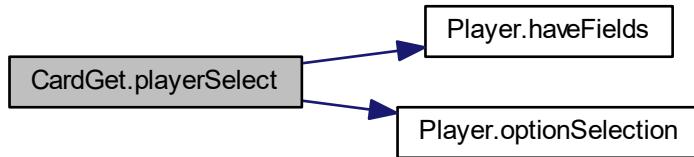
players size > 1 i algun de ells != jugador actual

Postcondició

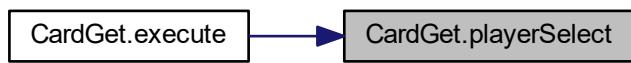
El jugador seleccionat ha estat retornat i té com a mínim un terreny o null si no hi ha cap jugador amb terrenys

Definició a la línia 47 del fitxer CardGet.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.11.3.4 `toString()`

```
String CardGet.toString( )
```

Sortida.

Precondició

true

Postcondició

el missatge ha estat retornat

Retorna

missatge de sortida

Definició a la línia 101 del fitxer CardGet.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [CardGet.java](#)

5.12 Referència de la Classe CardGive

Implementa la funcions de la carta de tipus donar.

Diagrama d'Herència per a CardGive:

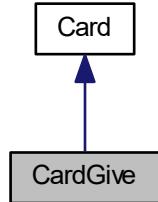
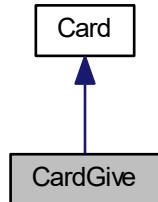


Diagrama de col·laboració per a CardGive:



Mètodes públics

- **CardGive** (boolean *postposable*)
Constructor de CardGive.
- void **execute** (ArrayList< Player > players, Player current_player)
Execució de la carta.
- String **toString** ()
Sortida.

Mètodes Privats

- Player **playerSelect** (ArrayList< Player > players, Player current_player)
Mostra tots els jugadors i deixa escollir a un d'ells a qui donar-li un terreny.
- BoxField **fieldSelect** (Player current_player)
Mostra tots els terreyns del jugador i deixa escollir a un d'ells.

5.12.1 Descripció Detallada

Implementa la funcions de la carta de tipus donar.

Definició a la línia 9 del fitxer CardGive.java.

5.12.2 Documentació del Constructor i el Destructor

5.12.2.1 CardGive()

```
CardGive.CardGive (
    boolean postposable )
```

Constructor de CardGive.

Precondició

true

Postcondició

La carta ha estat creada

Paràmetres

<i>postposable</i>	true si es posposable, false altrament
--------------------	--

Definició a la línia 17 del fitxer CardGive.java.

5.12.3 Documentació de les Funcions Membre

5.12.3.1 execute()

```
void CardGive.execute (
    ArrayList< Player > players,
    Player current_player )
```

Execució de la carta.

Precondició

true

Postcondició

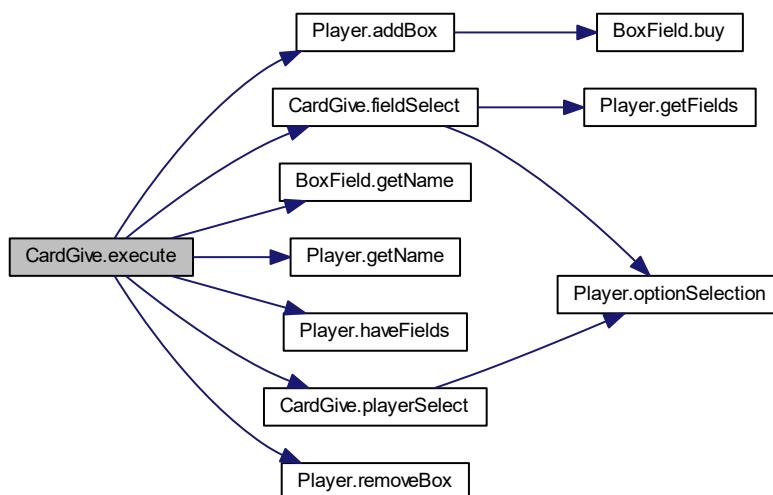
la propietat ha estat donada

Paràmetres

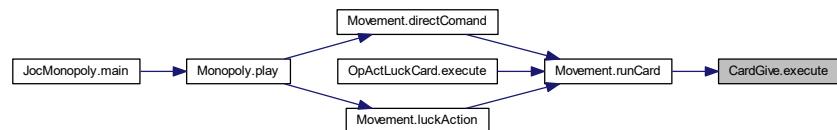
<i>players</i>	llista de jugadors de la partida
<i>current_player</i>	jugador actiu

Definició a la línia 28 del fitxer CardGive.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.12.3.2 fieldSelect()

```
BoxField CardGive.fieldSelect (
    Player current_player ) [private]
```

Mostra tots els terreys del jugador i deixa escollir a un d'ells.

Paràmetres

<code>current_player</code>	Jugador que ha de donar el terreny
-----------------------------	------------------------------------

Retorna

el terreny seleccionat per el jugador o null si no es pot

Precondició

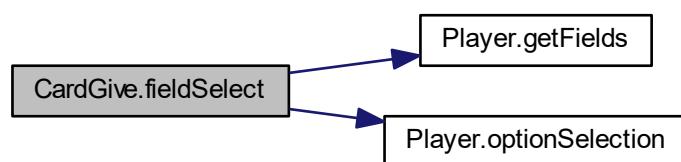
`current player fields > 0`

Postcondició

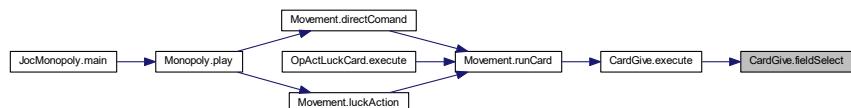
El terreny seleccionat ha estat retornat (mai retorna un terreny null per la precondició)

Definició a la línia 69 del fitxer CardGive.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.12.3.3 playerSelect()

```
Player CardGive.playerSelect (
    ArrayList< Player > players,
    Player current_player ) [private]
```

Mostra tots els jugadors i deixa escollir a un d'ells a qui donar-li un terreny.

Paràmetres

<i>players</i>	ArrayList de jugadors que estan jugant al Monopoly.
<i>current_player</i>	Jugador que ha de donar el terreny

Retorna

el jugador seleccionat o null si no n'hi ha cap

Precondició

players size > 1

Postcondició

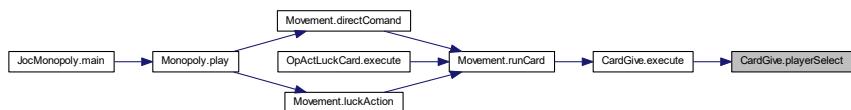
El jugador seleccionat ha estat retornat (mai retorna un jugador null per la precondició)

Definició a la línia 49 del fitxer CardGive.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.12.3.4 `toString()`

`String CardGive.toString ()`

Sortida.

Precondició

`true`

Postcondició

el missatge ha estat retornat

Retorna

missatge de sortida

Definició a la línia 93 del fitxer CardGive.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [CardGive.java](#)

5.13 Referència de la Classe CardGo

Targeta que mou al jugador a una casella determinada en el tauler de [Monopoly](#).

Diagrama d'Herència per a CardGo:

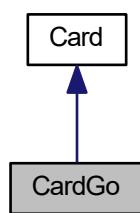
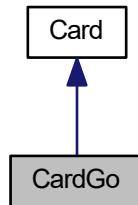


Diagrama de col·laboració per a CardGo:



Mètodes públics

- `CardGo (boolean postposable, int position)`
Constructor de CardGo.
- void `execute (Board board, Player current_player, ArrayList< String > rewards, OutputManager output)`
Mètode per executar la targeta CardGo. Moure el jugador fins a la posició que li indiqui la targeta.
- String `toString ()`
toString per mostrar la descripció de la targeta CardGo per text.

Atributs Privats

- final int `position`

5.13.1 Descripció Detallada

Targeta que mou al jugador a una casella determinada en el tauler de [Monopoly](#).

Definició a la línia 9 del fitxer CardGo.java.

5.13.2 Documentació del Constructor i el Destructor

5.13.2.1 CardGo()

```
CardGo.CardGo (
    boolean postposable,
    int position )
```

Constructor de [CardGo](#).

Precondició

```
true
```

Postcondició

Crea una targeta [CardGo](#) amb les propietats entrades i heredades.

Paràmetres

<i>postposable</i>	true si la targeta es pot posposar i ser guardada pel jugador, false altrament.
<i>position</i>	posició final on ha de moure's el jugador en cas d'executar-la.

Definició a la línia 19 del fitxer CardGo.java.

5.13.3 Documentació de les Funcions Membre

5.13.3.1 execute()

```
void CardGo.execute (
    Board board,
    Player current_player,
    ArrayList< String > rewards,
    OutputManager output )
```

Mètode per executar la targeta CardGo.Moura el jugador fins a la posició que li indiqui la targeta.

Precondició

players != null, board != null, current_player != null, rewards != null

Postcondició

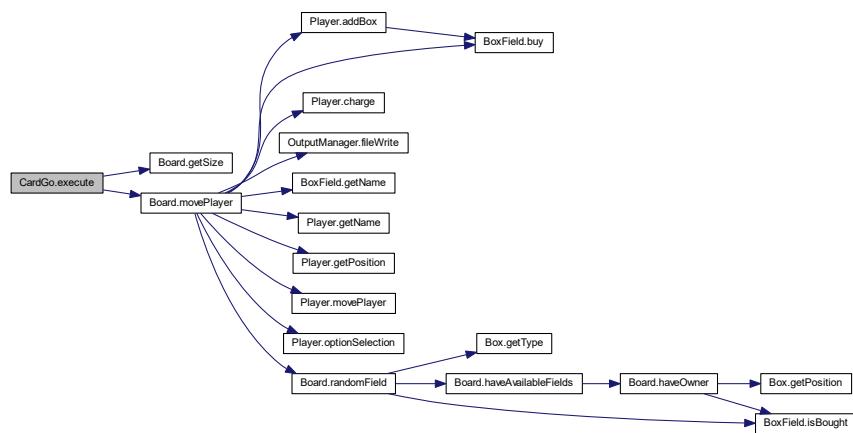
El jugador ha estat mogut fins a la posició position.

Paràmetres

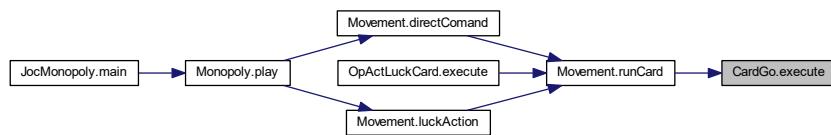
<i>board</i>	tauler de Monopoly en el qual s'està jugant.
<i>current_player</i>	jugador actual que executa la targeta CardGo.
<i>rewards</i>	ArrayList de recompenses que pot donar-te la casella de sortida.

Definició a la línia 32 del fitxer CardGo.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.13.3.2 `toString()`

```
String CardGo.toString ( )
```

`toString` per mostrar la descripció de la targeta `CardGo` per text.

Precondició

`true`

Postcondició

`CardGo` ha estat mostrada per pantalla.

Retorna

String de la sortida per pantalla.

Definició a la línia 47 del fitxer `CardGo.java`.

5.13.4 Documentació de les Dades Membre

5.13.4.1 position

```
final int CardGo.position [private]
```

Definició a la línia 10 del fitxer CardGo.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [CardGo.java](#)

5.14 Referència de la Classe CardPay

Targeta que fa pagar la quantitat assignada a un jugador.

Diagrama d'Herència per a CardPay:

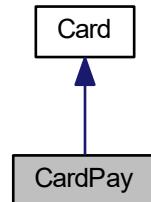
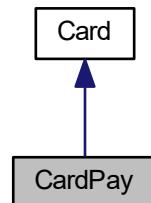


Diagrama de col·laboració per a CardPay:



Mètodes públics

- `CardPay` (boolean `postposable`, int `quantity`)
Constructor de CardPay.
- void `execute` (ArrayList<`Player`> `players`, `Board` `board`, `Player` `current_player`, `Movement` `aux`)
Execució de la carta.
- String `toString` ()
Sortida.

Atributs Privats

- final int `quantity`
quantitat a pagar

5.14.1 Descripció Detallada

Targeta que fa pagar la quantitat assignada a un jugador.

Definició a la línia 9 del fitxer CardPay.java.

5.14.2 Documentació del Constructor i el Destructor

5.14.2.1 CardPay()

```
CardPay.CardPay (
    boolean postposable,
    int quantity )
```

Constructor de `CardPay`.

Precondició

`true`

Postcondició

La carta ha estat creada

Paràmetres

<code>postposable</code>	true si es posposable, false altrament
<code>quantity</code>	quantitat a pagar

Definició a la línia 19 del fitxer CardPay.java.

5.14.3 Documentació de les Funcions Membre

5.14.3.1 execute()

```
void CardPay.execute (  
    ArrayList< Player > players,  
    Board board,  
    Player current_player,  
    Movement aux )
```

Execució de la carta.

Precondició

true

Postcondició

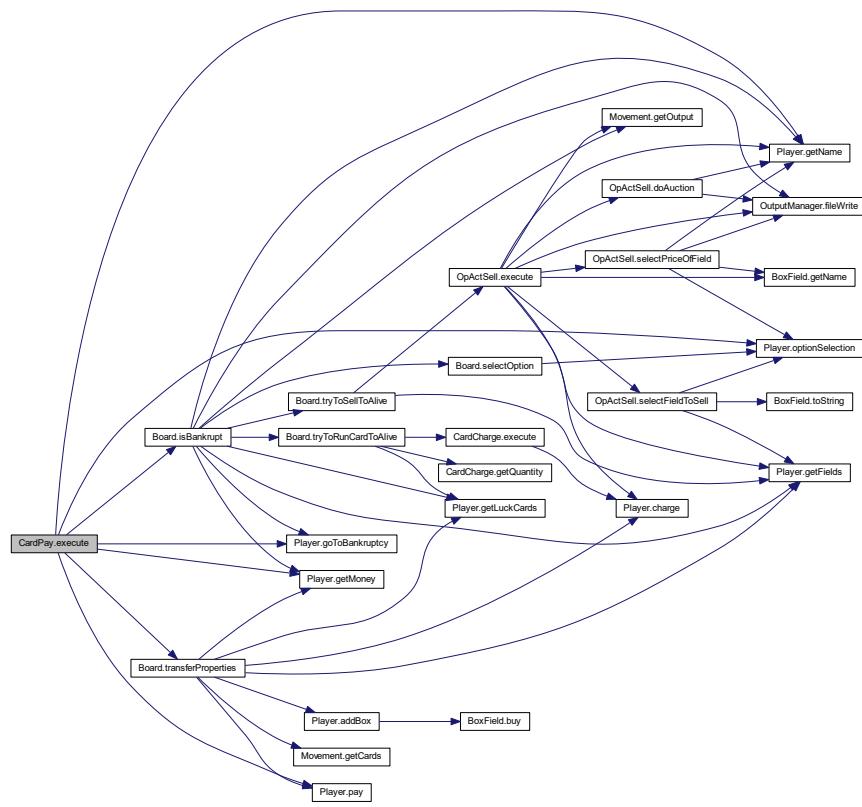
la quantitat ha estat pagada a un altre jugador

Paràmetres

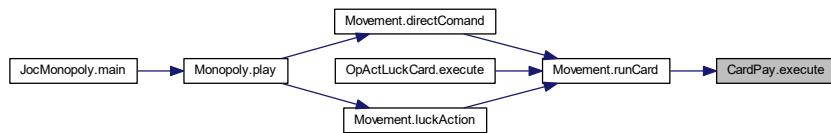
<i>players</i>	llista de jugadors de la partida
<i>board</i>	taulell
<i>current_player</i>	jugador actiu
<i>aux</i>	Classe movement per poder cridar a accions opcionals

Definició a la línia 33 del fitxer CardPay.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.14.3.2 `toString()`

```
String CardPay.toString ()
```

Sortida.

Precondició

true

Postcondició

el missatge ha estat retornat

Retorna

missatge de sortida

Definició a la línia 71 del fitxer CardPay.java.

5.14.4 Documentació de les Dades Membre

5.14.4.1 quantity

```
final int CardPay.quantity [private]
```

quantitat a pagar

Definició a la línia 10 del fitxer CardPay.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [CardPay.java](#)

5.15 Referència de la Classe CPUPlayer

Jugador controlat per el ordinador.

Diagrama d'Herència per a CPUPlayer:

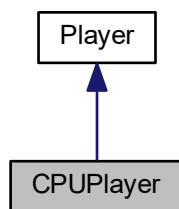
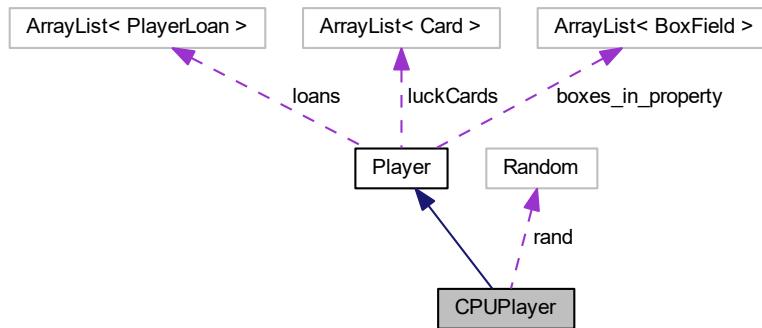


Diagrama de col·laboració per a CPUPlayer:



Mètodes públics

- **CPUPlayer** (String name, int initial_money, int initial_position)
Constructor de Player de tipus CPU.
- int **optionSelection** (String type, Player player, BoxField field, ArrayList< Integer > options, ArrayList< Player > players, Card card, int value, ArrayList< optionalActions > optional_actions)
Pregunta a la CPU segons el tipus entrat.
- String **stringValueSelection** (String type, Player player, BoxField field, int value, int second_value)
Pregunta a la CPU segons el tipus entrat.

Mètodes Privats

- int **optionalActionSelector** (Player player, ArrayList< optionalActions > optional_actions)
Selecciona la millor acció opcional per la CPU al final del torn segons les probabilitats de guanyar diners o terrenys.
- int **start** (int options)
Selecciona la millor acció de recompensa al passar per la casella de sortida.
- int **buy** (BoxField field)
Selecciona si compra o no el terreny entrat.
- int **buildChoice** (BoxField field)
Selecciona si construeix o no en el terreny.
- int **build** (BoxField field)
Selecciona si construeix un apartament o un hotel.
- int **buildApartment** (BoxField field, int max)
Selecciona quants apartament vol construir.
- int **betQuantity** ()
Selecciona la millor quantitat a apostar (el 10% del seu capital)
- int **betValue** (int betQuantity)
Selecciona la millor aposta a fer depenguent de la quantitat apostada.
- int **cardGetPlayerSelect** (ArrayList< Player > players)
Selecciona el jugador amb mes cartes.
- int **cardGetFieldSelect** ()
Selecciona el terreny amb major valor.

- int `cardGivePlayerSelect` (ArrayList< Player > players)
Selecciona el jugador amb mes cartes.
- int `cardGiveFieldSelect` ()
Selecciona el terreny amb major valor.
- int `cardPayPlayerSelect` (ArrayList< Player > players)
Selecciona el jugador a qui pagar.
- int `postposableLuckCardChoice` (Card card)
Selecciona si vol executar una carta de sort o guardar-sela.
- int `buyPlayerSelect` (ArrayList< Player > players)
Selecció del jugador a la funció Buy.
- int `buyFieldSelect` ()
Selecció del terreny a la funció Buy.
- int `buyInitialOffer` (BoxField field)
Selecció de la oferta inicial a la funció Buy.
- String `buyBuyerOffer` (BoxField field, int actual_offer)
Selecció de la oferta del comprador a la funció Buy en negociació
- String `buySellerOffer` (BoxField field, int actual_offer)
Selecció de la oferta del venedor a la funció Buy en negociació
- int `sellFieldSelect` ()
Selecció del terreny a la funció Sell.
- int `sellInitialOffer` (BoxField field)
Selecció de la oferta inicial a la funció Sell.
- int `sellBuyerOffer` (BoxField field, int actual_offer)
Selecció de la oferta del comprador a la funció Sell.
- int `loanPlayerSelect` (ArrayList< Player > players)
Selecció del jugador a la funció Loan.
- int `loanInitialOffer` ()
Selecció de la oferta inicial del prestatge.
- String `loanInterestOffer` (Player player, int offer, int interests)
Selecció de la acceptació, denegació o valor del interès del prestatge.
- String `loanTurnsOffer` ()
Selecció de dels torns que durarà del prestatge.
- int `bankruptcy` ()
Selecció de la opció a triar si s'entra en bancarrota (la CPU mai farà cap altre opció que declarar fallida perquè sino la partida pot durar eternament)

Atributs Privats

- final Random `rand` = new Random()

5.15.1 Descripció Detallada

Jugador controlat per el ordinador.

Definició a la línia 12 del fitxer CPUPlayer.java.

5.15.2 Documentació del Constructor i el Destructor

5.15.2.1 CPUPlayer()

```
CPUPlayer.CPUPlayer (
    String name,
    int initial_money,
    int initial_position )
```

Constructor de [Player](#) de tipus CPU.

Paràmetres

<i>name</i>	nom del Jugador.
<i>initial_money</i>	quantitat de diners inicials dels que disposa un Jugador.
<i>initial_position</i>	posició inicial del Jugador.

Precondició

true

Postcondició

Crea un jugador de tipus CPU amb els atributs entrats.

Definició a la línia 24 del fitxer CPUPlayer.java.

5.15.3 Documentació de les Funcions Membre

5.15.3.1 bankruptcy()

```
int CPUPlayer.bankruptcy ( ) [private]
```

Selecció de la opció a triar si s'entra en bancarrota (la CPU mai farà cap altre opció que declarar fallida perquè sino la partida pot durar eternament)

Precondició

true

Postcondició

la opció escollida s'ha retornat

Retorna

0

Reimplementat de [Player](#).

Definició a la línia 614 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:



5.15.3.2 betQuantity()

```
int CPUPlayer.betQuantity ( ) [private]
```

Selecciona la millor quantitat a apostar (el 10% del seu capital)

Precondició

true

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 282 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.3 betValue()

```
int CPUPlayer.betValue (
    int betQuantity ) [private]
```

Selecciona la millor aposta a fer depenguent de la quanitat apostada.

Paràmetres

<i>betQuantity</i>	quantitat apostada
--------------------	--------------------

Precondició

$\text{betQuantity} < \text{player money}$

Postcondició

la opció escollida s'ha retornat

Retorna

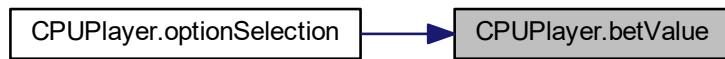
la opció escollida

Definició a la línia 293 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.4 build()

```
int CPUPlayer.build (
    BoxField field ) [private]
```

Selecciona si construeix un apartament o un hotel.

Paràmetres

<i>field</i>	Terreny a construir
--------------	---------------------

Precondició

player `money()` > field `priceToBuild` (es pot parmetre mínim un apartament)

Postcondició

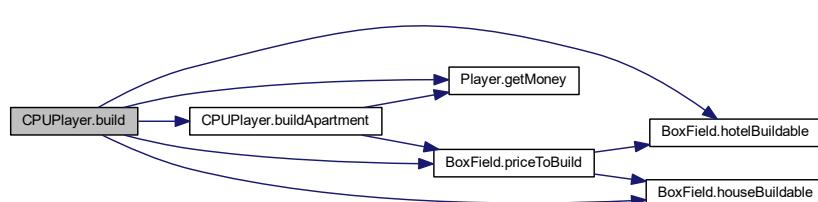
la opció escollida s'ha retornat

Retorna

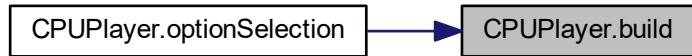
la opció escollida

Definició a la línia 257 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.5 buildApartment()

```

int CPUPlayer.buildApartment (
    BoxField field,
    int max ) [private]
  
```

Selecciona quants apartament vol construir.

Paràmetres

<i>field</i>	Terreny a contruir
<i>max</i>	numero maxim de apartaments permesos i asequibles per el jugador

Precondició

player `money()` > field `priceToBuild` (es pot parmetre minim un apartament)

Postcondició

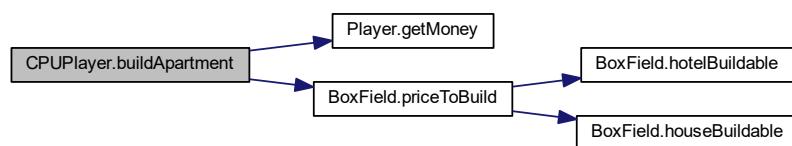
la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 271 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.6 buildChoice()

```
int CPUPlayer.buildChoice (
    BoxField field ) [private]
```

Selecciona si construeix o no en el terreny.

Paràmetres

<i>field</i>	Terreny a construir
--------------	---------------------

Precondició

field buildable == true

Postcondició

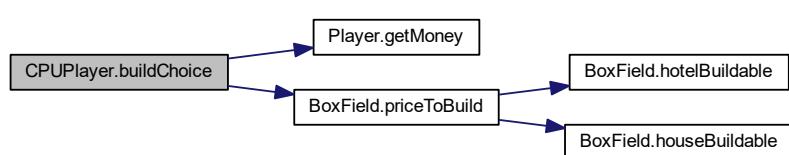
la opció escollida s'ha retornat

Retorna

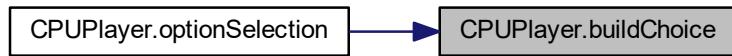
la opció escollida

Definició a la línia 245 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.7 buy()

```
int CPUPlayer.buy (
    BoxField field ) [private]
```

Selecciona si compra o no el terreny entrat.

Paràmetres

<i>field</i>	Terreny a comprar
--------------	-------------------

Precondició

true

Postcondició

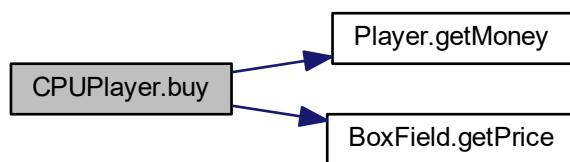
la opció escollida s'ha retornat

Retorna

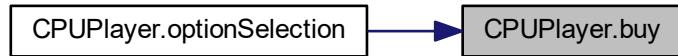
la opció escollida

Definició a la línia 233 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.8 buyBuyerOffer()

```

String CPUPlayer.buyBuyerOffer (
    BoxField field,
    int actual_offer ) [private]
  
```

Seleccio de la oferta del comprador a la funcio Buy en negociació

Paràmetres

<i>field</i>	terreny a comprar
<i>actual_offer</i>	ultima oferta activa

Precondició

true

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 470 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.9 buyFieldSelect()

`int CPUPlayer.buyFieldSelect () [private]`

Selecció del terreny a la funció Buy.

Precondició

`true`

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 434 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.10 buyInitialOffer()

```
int CPUPlayer.buyInitialOffer (
    BoxField field ) [private]
```

Selecció de la oferta inicial a la funció Buy.

Paràmetres

<i>field</i>	terreny a comprar
--------------	-------------------

Precondició

true

Postcondició

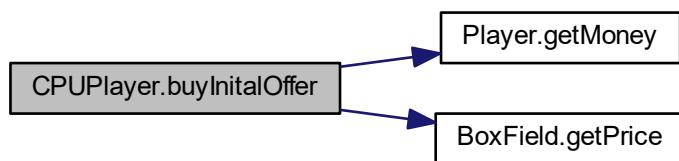
la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 453 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.11 buyPlayerSelect()

```
int CPUPlayer.buyPlayerSelect (
    ArrayList< Player > players ) [private]
```

Selecció del jugador a la funció Buy.

Paràmetres

<i>players</i>	llista de jugadors
----------------	--------------------

Precondició

players size > 0

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 416 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:



5.15.3.12 buySellerOffer()

```
String CPUPlayer.buySellerOffer (
    BoxField field,
    int actual_offer ) [private]
```

Selecció de la oferta del venedor a la funció Buy en negociació

Paràmetres

<i>field</i>	terreny a comprar
<i>actual_offer</i>	ultima oferta activa

Precondició

true

Postcondició

la opció escollida s'ha retornat

Retorna

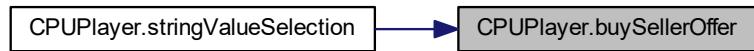
la opció escollida

Definició a la línia 487 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.15.3.13 cardGetFieldSelect()**

```
int CPUPlayer.cardGetFieldSelect ( ) [private]
```

Selecciona el terrent amb major valor.

Precondició

`field_owner fields size > 0` (El jugador té terrenys disponibles)

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 327 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.15.3.14 cardGetPlayerSelect()**

```

int CPUPlayer.cardGetPlayerSelect (
    ArrayList< Player > players ) [private]
  
```

Selecciona el jugador amb mes cartes.

Paràmetres

<i>players</i>	llista de jugadors de la partida
----------------	----------------------------------

Precondició

options size > 0

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 306 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:

**5.15.3.15 cardGiveFieldSelect()**

```
int CPUPlayer.cardGiveFieldSelect ( ) [private]
```

Selecciona el terreny amb major valor.

Precondició

field_owner fields size > 0 (El jugador té terrenys disponibles)

Postcondició

la opció escollida s'ha retornat

Retorna

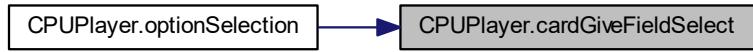
la opció escollida

Definició a la línia 361 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.16 cardGivePlayerSelect()

```
int CPUPlayer.cardGivePlayerSelect (
    ArrayList< Player > players ) [private]
```

Selecciona el jugador amb mes cartes.

Paràmetres

<i>players</i>	llista de jugadors de la partida
----------------	----------------------------------

Precondició

`options size > 0`

Postcondició

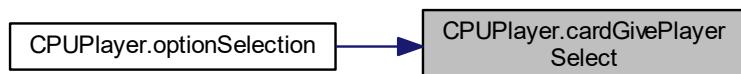
la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 343 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:



5.15.3.17 cardPayPlayerSelect()

```
int CPUPlayer.cardPayPlayerSelect (   
    ArrayList< Player > players ) [private]
```

Selecciona el jugador a qui pagar.

Paràmetres

<i>players</i>	llista de jugadors
----------------	--------------------

Precondició

players size > 1

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 380 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:



5.15.3.18 loanInitialOffer()

```
int CPUPlayer.loanInitialOffer ( ) [private]
```

Selecció de la oferta inicial adel prestec.

Precondició

true

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 570 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.15.3.19 loanInterestOffer()**

```

String CPUPlayer.loanInterestOffer (
    Player player,
    int offer,
    int interests )  [private]

```

Selecció de la acceptació, denegació o valor del interès del prestec.

Precondició

true

Postcondició

la opció escollida s'ha retornat

Retorna

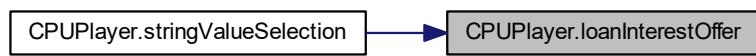
un enter amb el interes del prestec, no si rebutja la oferta o ok si la accepta

Definició a la línia 583 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.20 loanPlayerSelect()

```

int CPUPlayer.loanPlayerSelect (
    ArrayList< Player > players ) [private]
  
```

Selecció del jugador a la funció Loan.

Paràmetres

<i>players</i>	llista de jugadors
----------------	--------------------

Precondició

players size > 0

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 552 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:



5.15.3.21 loanTurnsOffer()

`String CPUplayer.loanTurnsOffer () [private]`

Selecció de dels torns que durarà del prestec.

Precondició

`true`

Postcondició

la opció escollida s'ha retornat

Retorna

numero de torns que durarà el prestec

Definició a la línia 603 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:



5.15.3.22 optionalActionSelector()

```

int CPUPlayer.optionalActionSelector (
    Player player,
    ArrayList< optionalActions > optional_actions ) [private]
  
```

Selecciona la millor acció opcional per la CPU al final del torn segons les probabilitats de guanyar diners o terrenys.

Paràmetres

<i>player</i>	Jugador actual
---------------	----------------

Precondició

true

Postcondició

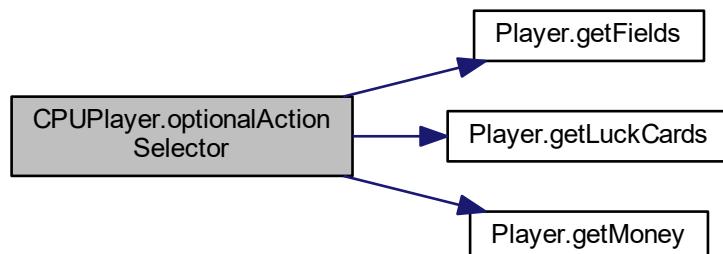
la opció escollida s'ha retornat

Retorna

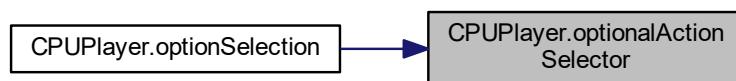
la opció escollida

Definició a la línia 163 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.23 optionSelection()

```
int CPUPlayer.optionSelection (
    String type,
    Player player,
    BoxField field,
    ArrayList< Integer > options,
    ArrayList< Player > players,
    Card card,
    int value,
    ArrayList< optionalActions > optional_actions )
```

Pregunta a la CPU segons el tipus entrat.

Paràmetres

<i>type</i>	tipus de pregunta a la CPU
<i>player</i>	possible jugador necessitat per la funció
<i>field</i>	possible terreny necessitat per la funció
<i>options</i>	llista de opcions possibles a triar
<i>players</i>	llista de jugadors de la partida
<i>card</i>	possible carta necessitada per la funció
<i>value</i>	possible valor enter necessitat per la funció

Precondició

type equival a una pregunta a la CPU (optionalActionSelector,start,buy,confirmation,buildChoice,build,build← Apartment,betQuantity,betQuantity,betValue,cardGetPlayerSelect,cardGetFieldSelect,cardGivePlayer← Select,cardGiveFieldSelect,ardPayPlayerSelect,postposableLuckCardChoice,buyPlayerSelect,buyField← Select,buyInitialOffer,sellFieldSelect,sellInitialOffer,sellBuyerOffer,loanPlayerSelect,loanInitialOffer,bankruptcy)

Postcondició

el valor escollit per la CPU s'ha retornat

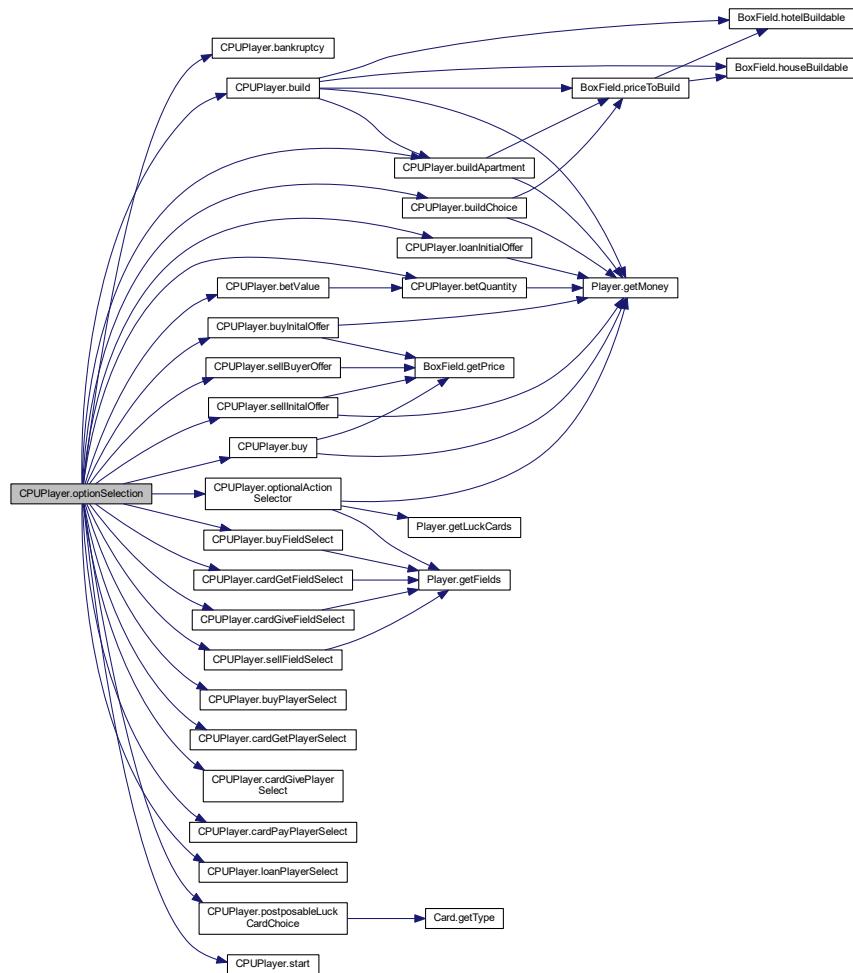
Retorna

el valor escollit per la CPU

Reimplementat de [Player](#).

Definició a la línia 41 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



5.15.3.24 postposableLuckCardChoice()

```
int CPUPlayer.postposableLuckCardChoice (
    Card card ) [private]
```

Selecciona si vol executar una carta de sort o guardar-sela.

Paràmetres

<code>card</code>	carta de sort
-------------------	---------------

Precondició

true

Postcondició

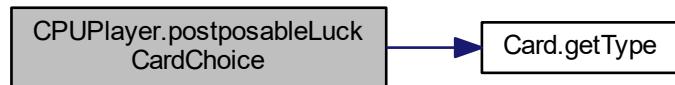
la opció escollida s'ha retornat

Retorna

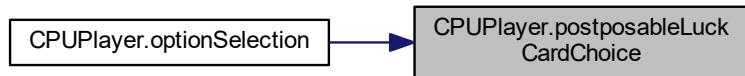
la opció escollida

Definició a la línia 404 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.15.3.25 sellBuyerOffer()**

```
int CPUPlayer.sellBuyerOffer (
    BoxField field,
    int actual_offer ) [private]
```

Selecció de la oferta del comprador a la funció Sell.

Paràmetres

<i>field</i>	terreny a comprar
<i>actual_offer</i>	ultima oferta activa

Precondició

true

Postcondició

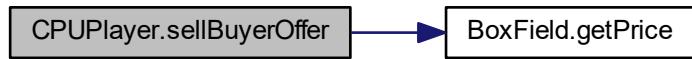
la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 537 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.15.3.26 sellFieldSelect()**

```
int CPUPlayer.sellFieldSelect ( ) [private]
```

Selecció del terreny a la funció Sell.

Precondició

true

Postcondició

la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 502 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.15.3.27 sellInitialOffer()**

```
int CPUPlayer.sellInitialOffer (
    BoxField field ) [private]
```

Selecció de la oferta inicial a la funció Sell.

Paràmetres

<i>field</i>	terreny a vendre
--------------	------------------

Precondició

true

Postcondició

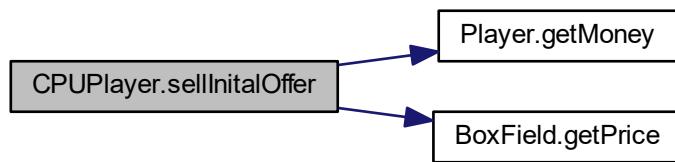
la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 521 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.15.3.28 start()

```
int CPUPlayer.start (
    int options ) [private]
```

Selecciona la millor acció de recompensa al passar per la casella de sortida.

Paràmetres

<i>options</i>	quantitat de recompenses possibles -1
----------------	---------------------------------------

Precondició

options es igual a la quantitat de recompenses possibles -1

Postcondició

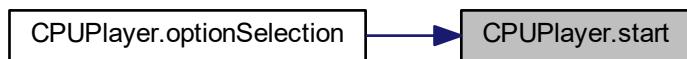
la opció escollida s'ha retornat

Retorna

la opció escollida

Definició a la línia 222 del fitxer CPUPlayer.java.

Gràfic de crides a aquesta funció:

**5.15.3.29 stringValueSelection()**

```

String CPUPlayer.stringValueSelection (
    String type,
    Player player,
    BoxField field,
    int value,
    int second_value )
  
```

Pregunta a la CPU segons el tipus entrat.

Paràmetres

<i>type</i>	tipus de pregunta a la CPU
<i>player</i>	possible jugador necessitat per la funcio
<i>field</i>	possible terreny necessitat per la funcio
<i>value</i>	possible enter necessitat per la funcio
<i>second_value</i>	possible enter necessitat per la funcio

Precondició

type equival a una pregunta a la CPU (throwDice,buyBuyerOffer,buySellerOffer,loanInterestOffer,loanTurnsOffer)

Postcondició

el string escollit per la CPU s'ha retornat

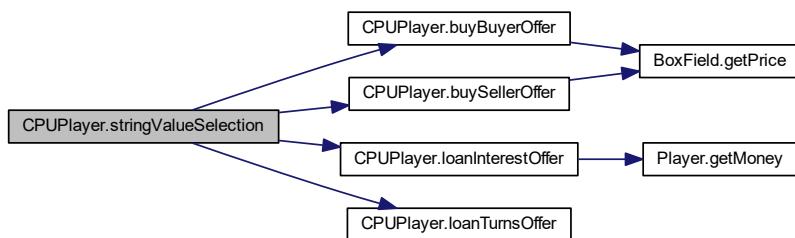
Retorna

el string escollit per la CPU

Reimplementat de [Player](#).

Definició a la línia 132 del fitxer CPUPlayer.java.

Gràfic de crides d'aquesta funció:



5.15.4 Documentació de les Dades Membre

5.15.4.1 rand

```
final Random CPUPlayer.rand = new Random() [private]
```

Definició a la línia 14 del fitxer CPUPlayer.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [CPUPlayer.java](#)

5.16 Referència de la Classe JocMonopoly

Main principal del joc del [Monopoly](#).

Mètodes Pùblics Estàtics

- static void [main](#) (String[] args)

Main principal per començar el Joc de [Monopoly](#).

Mètodes Privats Estàtics

- static void [printHelp \(\)](#)

5.16.1 Descripció Detallada

Main principal del joc del [Monopoly](#).

Definició a la línia 7 del fitxer JocMonopoly.java.

5.16.2 Documentació de les Funcions Membre

5.16.2.1 `main()`

```
static void JocMonopoly.main (
    String[] args ) [static]
```

Main principal per començar el Joc de [Monopoly](#).

Precondició

true

Postcondició

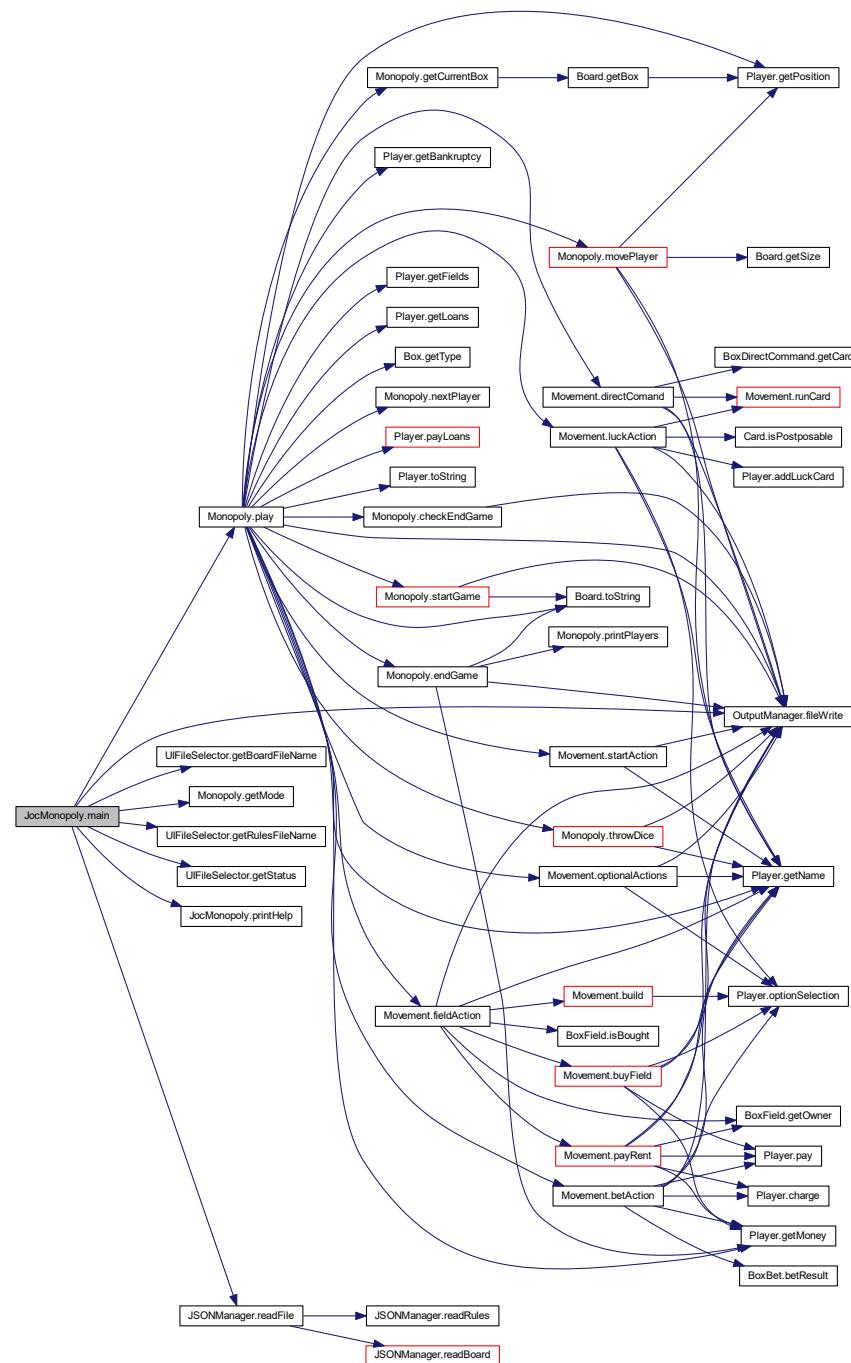
Joc acabat.

Paràmetres

<i>args</i>	fitxers d'entrada
-------------	-------------------

Definició a la línia 15 del fitxer JocMonopoly.java.

Gràfic de crides d'aquesta funció:



5.16.2.2 printHelp()

```
static void JocMonopoly.printHelp () [static], [private]
```

Definició a la línia 54 del fitxer JocMonopoly.java.

Gràfic de crides a aquesta funció:



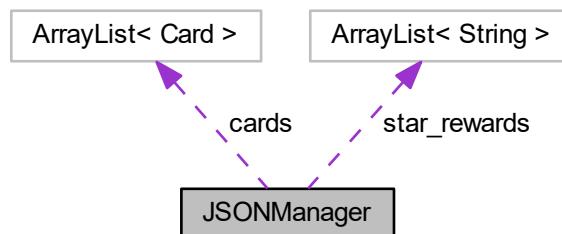
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [JocMonopoly.java](#)

5.17 Referència de la Classe JSONManager

[JSONManager](#) administra totes les funcions relacionades amb llegir fitxers JSON i també la creació del fitxer de desenvolupament de la partida.

Diagrama de col·laboració per a JSONManager:



Mètodes públics

- [JSONManager](#) (String rules, String board)
Constructor de JSONManager.
- [Monopoly readFile \(\)](#)
Llegeix els dos fitxers d'entrada rules_file i board_file per poder crear el joc del Monopoly.

Mètodes Privats

- [ArrayList<optionalActions> readRules \(\)](#)
Llegeix rules_file fitxer JSON i configura les propietats llegides del joc del Monopoly.
- [Board readBoard \(\)](#)
Llegeix bord_file fitxer JSON i configura les propietats del tauler llegides al joc del Monopoly.

Atributs Privats

- String `mode`
- ArrayList< String > `star_rewards` = new ArrayList<>()

Llistat de recompenses que pot donar Start.
- ArrayList< Card > `cards` = new ArrayList<>()

Llistat de targetes sort de Monopoly.
- int `initial_money` = 0

Diners inicials de cada Jugador.
- final String `rules_file`

Fitxer definició regles del joc de Monopoly.
- final String `board_file`

Fitxer definició tauler del joc de Monopoly.

5.17.1 Descripció Detallada

`JSONManager` administra totes les funcions relacionades amb llegir fitxers JSON i també la creació del fitxer de desenvolupament de la partida.

Autor

Gerard Rovellat

Definició a la línia 12 del fitxer `JSONManager.java`.

5.17.2 Documentació del Constructor i el Destructor

5.17.2.1 `JSONManager()`

```
JSONManager(JSONManager (
    String rules,
    String board )
```

Constructor de `JSONManager`.

Precondició

true

Postcondició

Crea la classe `JSONManager` amb el nom dels fitxers d'entrada JSON.

Paràmetres

<code>rules</code>	nom del fitxer JSON el qual descriu les regles del joc.
<code>board</code>	nom del fitxer JSON el qual descriu el tauler i les seves propietats.

Definició a la línia 28 del fitxer JSONManager.java.

5.17.3 Documentació de les Funcions Membre

5.17.3.1 readBoard()

```
Board JSONManager.readBoard ( ) [private]
```

Llegeix bord_file fitxer JSON i configura les propietats del tauler llegides al joc del Monopoly.

Precondició

true

Postcondició

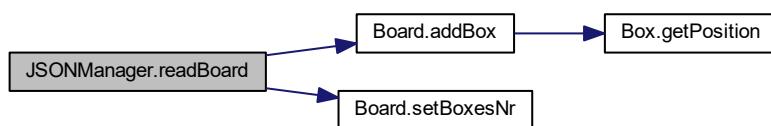
board_file ha estat llegit.

Retorna

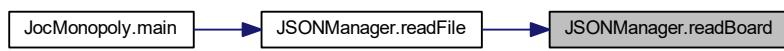
Tauler amb totes les seves propietats.

Definició a la línia 119 del fitxer JSONManager.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.17.3.2 readFile()

`Monopoly JSONManager.readFile ()`

Llegeix els dos fitxers d'entrada `rules_file` i `board_file` per poder crear el joc del [Monopoly](#).

Precondició

`true`

Postcondició

Retorna el joc del [Monopoly](#) amb les configuracions de `rules_file` i `board_file`.

Retorna

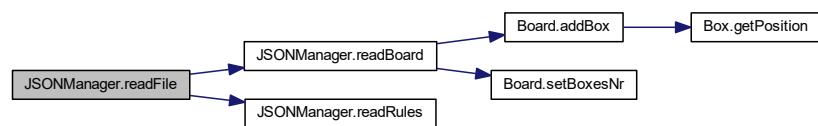
[Monopoly](#) `monopoly` amb totes les característiques establertes.

< Tauler llegit del fitxer JSON pel [Monopoly](#).

< Llistat d'accions opcionals possibles.

Definició a la línia 39 del fitxer `JSONManager.java`.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.17.3.3 `readRules()`

```
ArrayList<optionalActions> JSONManager.readRules () [private]
```

Llegeix `rules_file` fitxer JSON i configura les propietats llegides del joc del [Monopoly](#).

Precondició

`true`

Postcondició

`rules_file` ha estat llegit.

Retorna

`ArrayList` amb totes les accions opcionals possibles que el jugador pot fer en aquest joc.

Definició a la línia 53 del fitxer `JSONManager.java`.

Gràfic de crides a aquesta funció:



5.17.4 Documentació de les Dades Membre

5.17.4.1 `board_file`

```
final String JSONManager.board_file [private]
```

Fitxer definició tauler del joc de [Monopoly](#).

Definició a la línia 19 del fitxer `JSONManager.java`.

5.17.4.2 `cards`

```
ArrayList<Card> JSONManager.cards = new ArrayList<>() [private]
```

Llistat de targetes sort de [Monopoly](#).

Definició a la línia 16 del fitxer `JSONManager.java`.

5.17.4.3 initial_money

```
int JSONManager.initial_money = 0 [private]
```

Diners inicials de cada Jugador.

Definició a la línia 17 del fitxer JSONManager.java.

5.17.4.4 mode

```
String JSONManager.mode [private]
```

Definició a la línia 14 del fitxer JSONManager.java.

5.17.4.5 rules_file

```
final String JSONManager.rules_file [private]
```

Fitxer definicio regles del joc de [Monopoly](#).

Definició a la línia 18 del fitxer JSONManager.java.

5.17.4.6 star_rewards

```
ArrayList<String> JSONManager.star_rewards = new ArrayList<>() [private]
```

Llistat de recompenses que pot donar Start.

Definició a la línia 15 del fitxer JSONManager.java.

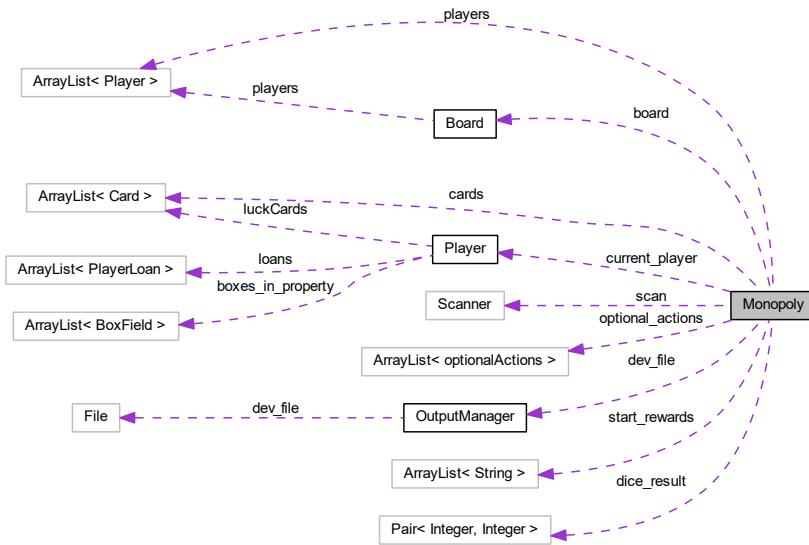
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [JSONManager.java](#)

5.18 Referència de la Classe Monopoly

Classe que s'encarrega d'administrar l'inicialització, els torns i funcionament general i la finalització del joc Monopoly.

Diagrama de col·laboració per a Monopoly:



Mètodes públics

- **Monopoly** (String mode, Board read_board, ArrayList<optionalActions> read_optional_actions, ArrayList<Card> stack_of_cards, int initial_money, ArrayList<String> start_rewards)

Constructor de Monopoly.
- void **play** ()

Mètode que s'encarrega de gestionar els fluxes de joc del Monopoly (comprovar si s'ha de seguir, jugar, finalitzar).
- String **getMode** ()

Retorna el mode de joc.

Mètodes Privats

- void **movePlayer** ()

Inicialitza l'acció de moure Jugador i dona el resultat dels daus.
- Box **getCurrentBox** ()

Retornara la casella actual del Jugador.
- Boolean **checkEndGame** ()

Comproba si el joc s'ha acabat o no.
- void **nextPlayer** ()

Salta al següent Jugador.
- void **throwDice** ()

Tira els daus d'un torn.
- void **startGame** ()

- **Inicialitza el joc del *Monopoly*.**
- int **askNrOfPlayers ()**

Demana el numero number_of_players de Jugadors que jugaran al joc del monopoly, han de jugar de dos a dotze persones.
- void **askInfoOfPlayers (int number_of_players)**

Demana la informació que es necessita dels Jugadors, en aquest cas, el seu nom i si es CPU o Usuari.
- void **endGame ()**

*Gestiona la finalització del joc de *Monopoly*.*
- void **printPlayers ()**

*Mostra els jugadors que no estan en fallida jugant a *Monopoly*.*

Atributs Privats

- final String **mode**

Mode de joc.
- final ArrayList< Player > **players** = new ArrayList<>()

*Llista de Jugadors del *Monopoly*.*
- final Board **board**

*Tauler del *Monopoly*.*
- final int **initial_money**

Diners inicials quan es comença una partida.
- final ArrayList< String > **start_rewards**

Recompenses de casella sortida "Start".
- Pair< Integer, Integer > **dice_result**

Resultat dels daus tirats.
- int **current_player_iterator** = 0

Iterador que recorre Llista players.
- Player **current_player**

Jugador actual.
- final ArrayList< optionalActions > **optional_actions**

Llista d'accions opcionals.
- final ArrayList< Card > **cards**

Llista de targetes sort.

5.18.1 Descripció Detallada

Classe que s'encarrega d'administrar l'inicialització, els torns i funcionament general i la finalització del joc *Monopoly*.

Definició a la línia 16 del fitxer Monopoly.java.

5.18.2 Documentació del Constructor i el Destructor

5.18.2.1 Monopoly()

```
Monopoly.Monopoly (
    String mode,
    Board read_board,
    ArrayList< optionalActions > read_optional_actions,
    ArrayList< Card > stack_of_cards,
    int initial_money,
    ArrayList< String > start_rewards )
```

Constructor de [Monopoly](#).

Precondició

true

Postcondició

Crea [Monopoly](#) amb els atributs entrats.

Paràmetres

<i>read_board</i>	tauler llegit del JSONManager .
<i>read_optional_actions</i>	accions opcionals llegides del JSONManager .
<i>stack_of_cards</i>	pila de targetes sort del Monopoly .
<i>initial_money</i>	diners inicials del Jugador.
<i>start_rewards</i>	llista de recompenses per la casella de sortida "Start"

Definició a la línia 42 del fitxer Monopoly.java.

5.18.3 Documentació de les Funcions Membre

5.18.3.1 askInfoOfPlayers()

```
void Monopoly.askInfoOfPlayers (
    int number_of_players ) [private]
```

Demana la informació que es necessita dels Jugadors, en aquest cas, el seu nom i si es CPU o Usuari.

Precondició

true

Postcondició

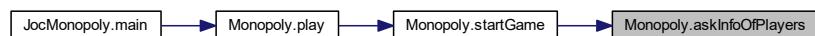
Els Jugadors entrats estan preparats per poder jugar al [Monopoly](#).

Definició a la línia 261 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.3.2 askNrOfPlayers()

```
int Monopoly.askNrOfPlayers ( ) [private]
```

Demana el numero `number_of_players` de Jugadors que jugaran al joc del monopoly, han de jugar de dos a dotze persones.

Precondició

`true`

Postcondició

Retorna el numero entrat de Jugadors que jugaran al [Monopoly](#).

Retorna

numero de Jugadors de la partida.

Definició a la línia 237 del fitxer Monopoly.java.

Gràfic de crides a aquesta funció:



5.18.3.3 checkEndGame()

```
Boolean Monopoly.checkEndGame ( ) [private]
```

Comproba si el joc s'ha acabat o no.

Precondició

true

Postcondició

Retorna true si el joc s'ha acabat false altrament.

Retorna

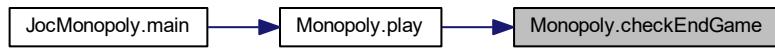
true si s'ha acabat, false altrament.

Definició a la línia 157 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.3.4 endGame()

```
void Monopoly.endGame ( ) [private]
```

Gestiona la finalització del joc de [Monopoly](#).

Precondició

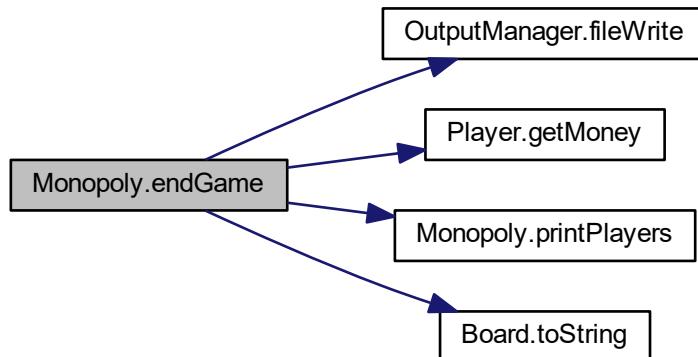
true

Postcondició

Finalitza el joc i mostra per pantalla el resum de la partida.

Definició a la línia 302 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.3.5 getCurrentBox()

```
Box Monopoly.getCurrentBox ( ) [private]
```

Retornara la casella actual del Jugador.

Precondició

true

Postcondició

Casella actual on esta el Jugador retornada.

Retorna

box on es el Jugador.

Definició a la línia 147 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.3.6 getMode()

```
String Monopoly.getMode ( )
```

Retorna el mode de joc.

Precondició

true

Postcondició

el mode de joc s'ha retornat.

Retorna

mode de joc

Definició a la línia 119 del fitxer Monopoly.java.

Gràfic de crides a aquesta funció:



5.18.3.7 movePlayer()

```
void Monopoly.movePlayer ( ) [private]
```

Inicialitza l'acció de moure Jugador i dona el resultat dels daus.

Precondició

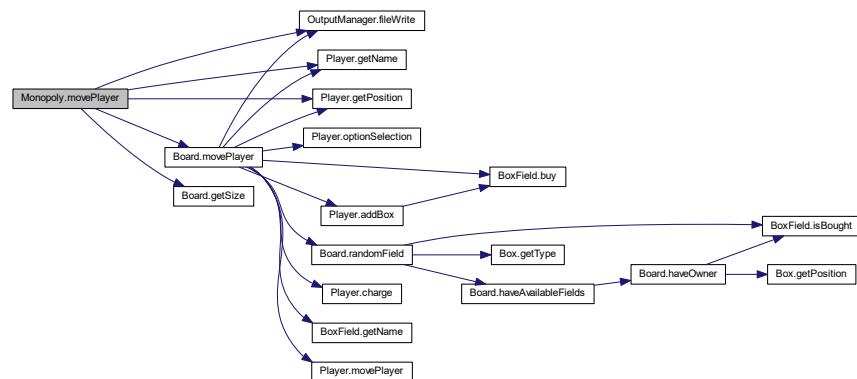
true

Postcondició

Jugador sera mogut fins a position del \board de [Monopoly](#).

Definició a la línia 128 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.3.8 nextPlayer()

```
void Monopoly.nextPlayer ( ) [private]
```

Salta al següent Jugador.

Precondició

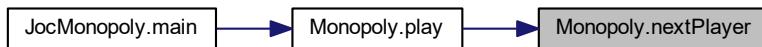
`true`

Postcondició

S'ha passat al següent Jugador de la llista.

Definició a la línia 181 del fitxer Monopoly.java.

Gràfic de crides a aquesta funció:

**5.18.3.9 play()**

```
void Monopoly.play ( )
```

Mètode que s'encarrega de gestionar els fluxe de joc del [Monopoly](#) (comprovar si s'ha de seguir, jugar, finalitzar).

Precondició

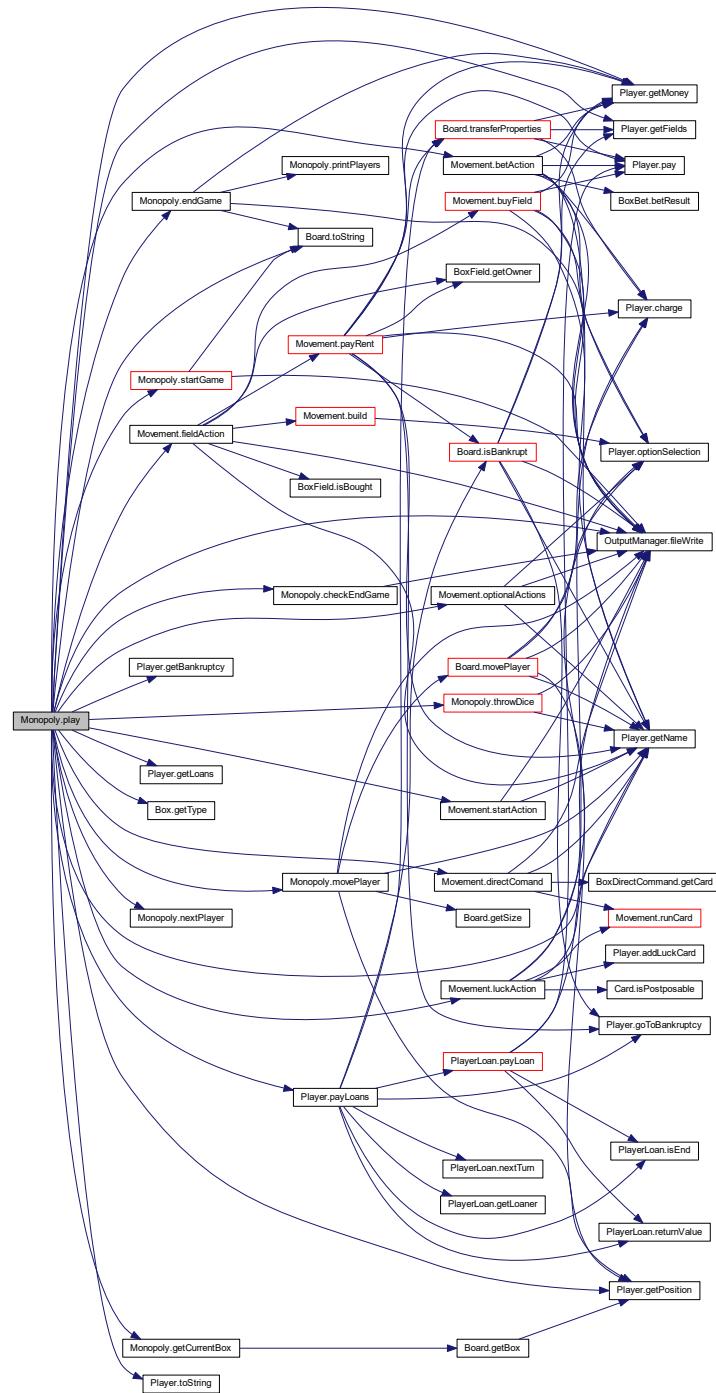
true

Postcondició

Es jugara al [Monopoly](#) fins que s'hagi acabat el joc.

Definició a la línia 57 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.3.10 printPlayers()

```
void Monopoly.printPlayers () [private]
```

Mostra els jugadors que no estan en fallida jugant a [Monopoly](#).

Precondició

```
true
```

Postcondició

Retorna els jugadors amb `bankruptcy = false`.

Definició a la línia 349 del fitxer `Monopoly.java`.

Gràfic de crides a aquesta funció:



5.18.3.11 startGame()

```
void Monopoly.startGame () [private]
```

Inicialitza el joc del [Monopoly](#).

Precondició

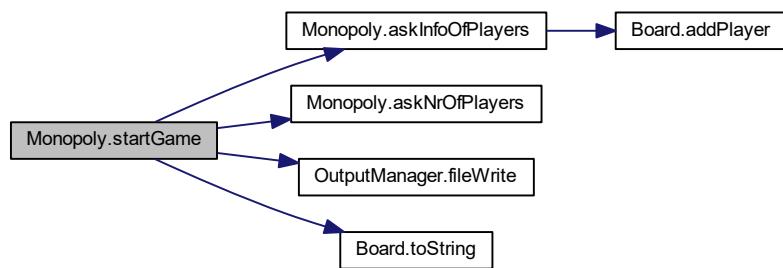
```
true
```

Postcondició

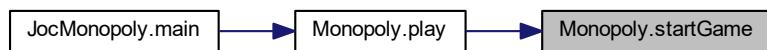
El joc ha estat inicialitzat i està preparat per començar la partida.

Definició a la línia 222 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.3.12 `throwDice()`

```
void Monopoly.throwDice( ) [private]
```

Tira els daus d'un torn.

Precondició

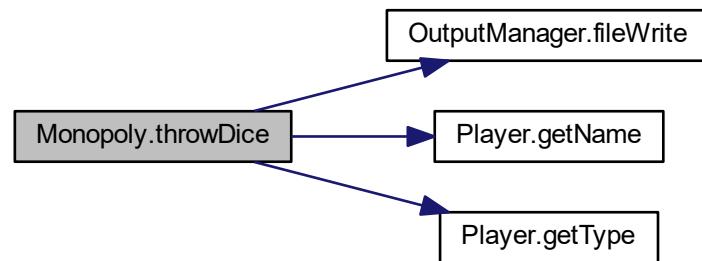
`true`

Postcondició

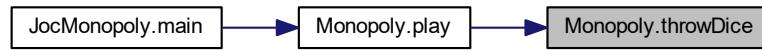
Dona el resultat dels daus d'aquest torn.

Definició a la línia 204 del fitxer Monopoly.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.18.4 Documentació de les Dades Membre

5.18.4.1 board

```
final Board Monopoly.board [private]
```

Tauler del [Monopoly](#).

Definició a la línia 20 del fitxer Monopoly.java.

5.18.4.2 cards

```
final ArrayList<Card> Monopoly.cards [private]
```

Llista de targetes sort.

Definició a la línia 27 del fitxer Monopoly.java.

5.18.4.3 current_player

```
Player Monopoly.current_player [private]
```

Jugador actual.

Definició a la línia 25 del fitxer Monopoly.java.

5.18.4.4 current_player_iterator

```
int Monopoly.current_player_iterator = 0 [private]
```

Iterador que recorre Llista players.

Definició a la línia 24 del fitxer Monopoly.java.

5.18.4.5 dice_result

```
Pair<Integer, Integer> Monopoly.dice_result [private]
```

Resultat dels daus tirats.

Definició a la línia 23 del fitxer Monopoly.java.

5.18.4.6 initial_money

```
final int Monopoly.initial_money [private]
```

Diners inicials quan es comença una partida.

Definició a la línia 21 del fitxer Monopoly.java.

5.18.4.7 mode

```
final String Monopoly.mode [private]
```

Mode de joc.

Definició a la línia 18 del fitxer Monopoly.java.

5.18.4.8 optional_actions

```
final ArrayList<optionalActions> Monopoly.optional_actions [private]
```

Llista d'accions opcionals.

Definició a la línia 26 del fitxer Monopoly.java.

5.18.4.9 players

```
final ArrayList<Player> Monopoly.players = new ArrayList<>() [private]
```

Llista de Jugadors del [Monopoly](#).

Definició a la línia 19 del fitxer Monopoly.java.

5.18.4.10 start_rewards

```
final ArrayList<String> Monopoly.start_rewards [private]
```

Recompenses de casella sortida "Start".

Definició a la línia 22 del fitxer Monopoly.java.

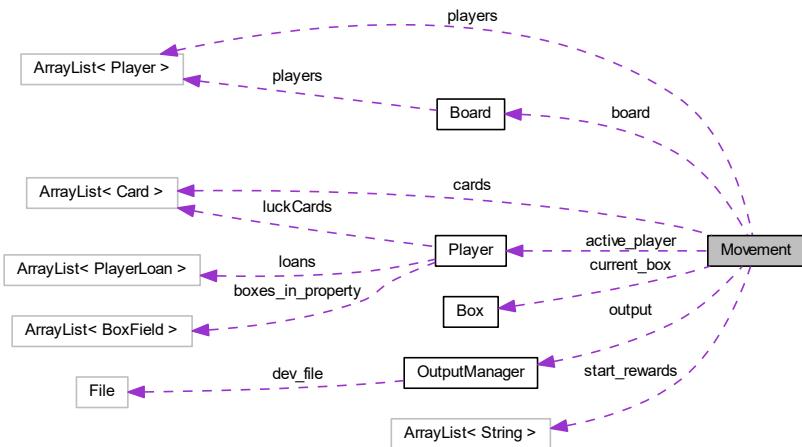
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [Monopoly.java](#)

5.19 Referència de la Classe Movement

Classe que s'encarrega de gestionar les accions que es poden fer en un moviment. Aquest moviment sera fet sobre un Jugador.

Diagrama de col·laboració per a Movement:



Mètodes públics

- **Movement** (**Box** box, **Player** player, **ArrayList< Player >** players, **Board** board, **ArrayList< String >** start_rewards, **ArrayList< Card >** cards, **OutputManager** output)

Constructor de **Movement**.
- void **startAction** ()

Informa al Jugador que ha caigut a la casella de sortida Start.
- void **fieldAction** ()

Administra les accions que s'han de fer si es cau en una Casella de terreny Field.
- void **betAction** ()

Administra les accions que s'han de fer si es cau en una Casella d'aposta Bet.
- void **luckAction** ()

Administra les accions que s'han de fer si es cau en una Casella de sort.
- void **directComand** ()

Administra les accions que s'han de fer si es cau en una Casella de comanda directa.
- void **runCard** (**Card** card)

Administra les accions que s'han de fer dependent de la targeta que et toqui.
- void **optionalActions** (**ArrayList< optionalActions >** possible_actions)

Administra totes les possibles accions opcionals.
- void **buyField** ()

Administrerà la compra d'un terreny on s'ha caigut de tauler.
- void **payRent** ()

Administrerà el pagament dels lloguers en els terrenys amb propietaris.
- void **build** ()

Administra l'acció per contruir edificis si el terreny es de la propietat de Jugador.
- **ArrayList< Card >** **getCards** ()

Retorna les targetes sort del joc del **Monopoly**.
- **OutputManager** **getOutput** ()

Retorna la classe de sortida de dades.

Mètodes Privats

- void **buildApartment** (**BoxField** field)
Administra l'acció per construir apartaments en una propietat.
- void **buildHotel** (**BoxField** field)
Administra l'acció per construir un hotel en una propietat.

Atributs Privats

- final **Box current_box**
Casella actual.
- final **Player active_player**
Jugador que està realitzant el moviment.
- final **ArrayList< String > start_rewards**
Llista de recompenses de la casella Sortida.
- final **ArrayList< Card > cards**
Llista de targetes sort de Monopoly.
- final **OutputManager output**

5.19.1 Descripció Detallada

Classe que s'encarrega de gestionar les accions que es poden fer en un moviment. Aquest moviment sera fet sobre un Jugador.

Definició a la línia 11 del fitxer Movement.java.

5.19.2 Documentació del Constructor i el Destructor

5.19.2.1 Movement()

```
Movement.Movement (
    Box box,
    Player player,
    ArrayList< Player > players,
    Board board,
    ArrayList< String > start_rewards,
    ArrayList< Card > cards,
    OutputManager output )
```

Constructor de **Movement**.

Precondició

```
current_posicion != null & active_player != null
```

Postcondició

Crea un moviment **Movement** amb **box** i **player**.

Paràmetres

<i>box</i>	Casella posició de <code>active_player</code> en el tauler.
<i>player</i>	Informació de el Jugador que està jugant en aquest torn.

Definició a la línia 28 del fitxer Movement.java.

5.19.3 Documentació de les Funcions Membre

5.19.3.1 betAction()

```
void Movement.betAction ( )
```

Administra les accions que s'han de fer si es cau en una Casella d'aposta Bet.

Precondició

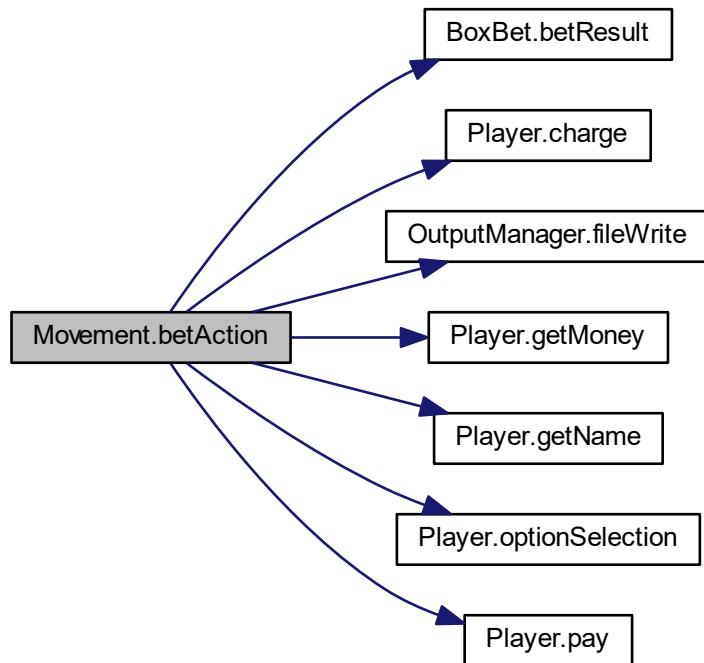
true

Postcondició

Dona la suma de diners de l'aposta que el jugador que fa el moviment ha realitzat.

Definició a la línia 68 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.2 build()

```
void Movement.build( )
```

Administra l'acció per contruir edificis si el terreny es de la propietat de Jugador.

Precondició

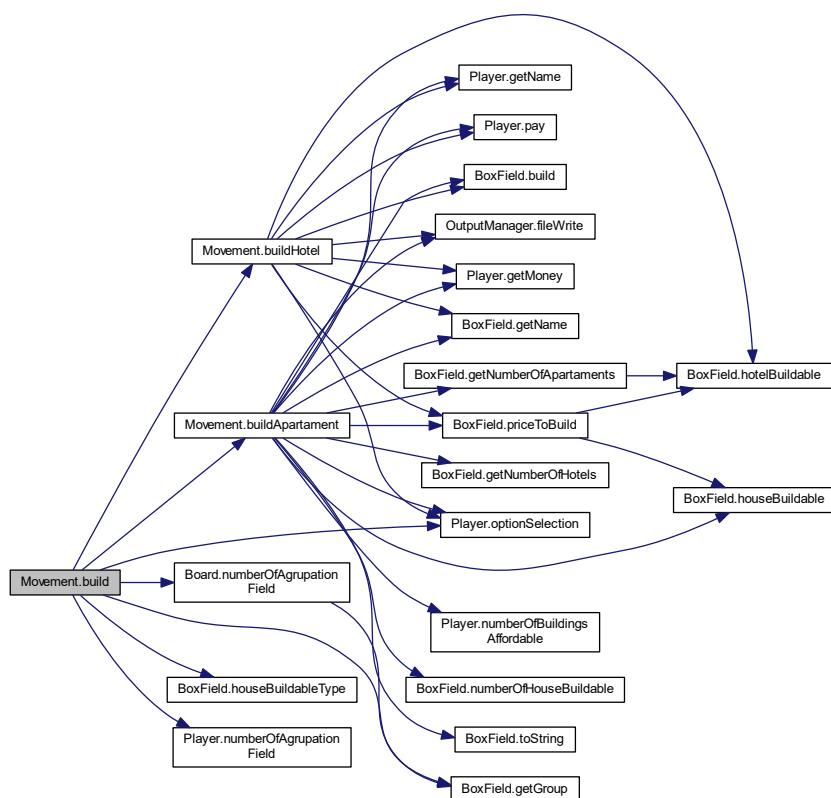
true

Postcondició

L'acció per contruir edificis dins d'una propietat ha estat gestionada.

Definició a la línia 308 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.3 buildApartament()

```
void Movement.buildApartament (   
    BoxField field ) [private]
```

Administra l'acció per contruir apartaments en una propietat.

Precondició

field != null

Postcondició

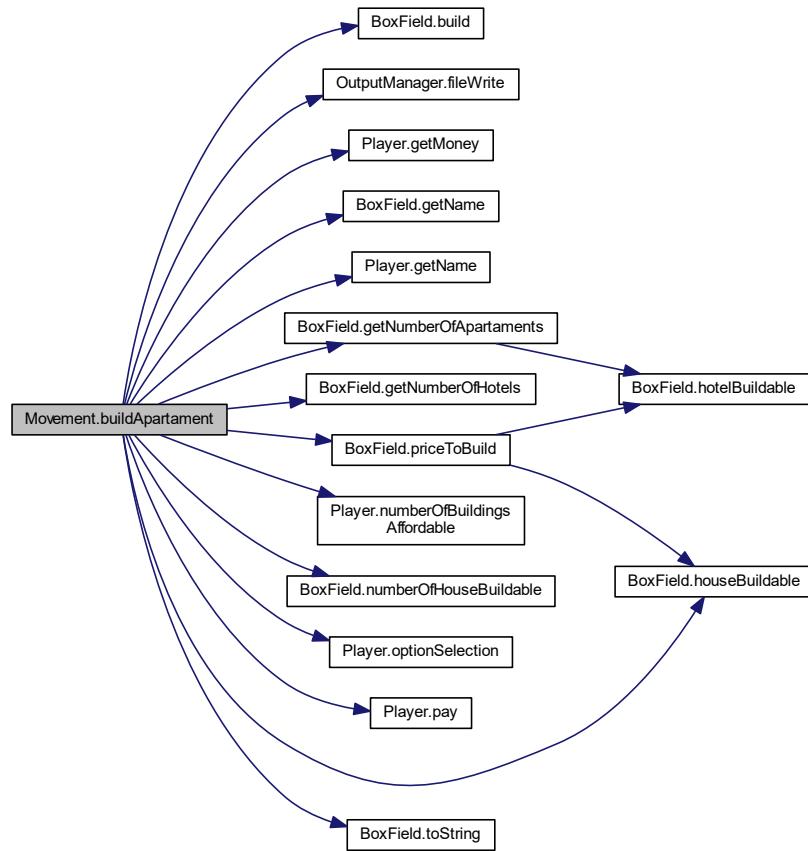
L'acció per contruir apartaments dins d'una propietat ha estat gestionada.

Paràmetres

<i>field</i>	terreny on s'ha de contruir apartaments.
--------------	--

Definició a la línia 354 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.4 buildHotel()

```
void Movement.buildHotel (
    BoxField field ) [private]
```

Administra l'acció per contruir un hotel en una propietat.

Precondició

`field != null`

Postcondició

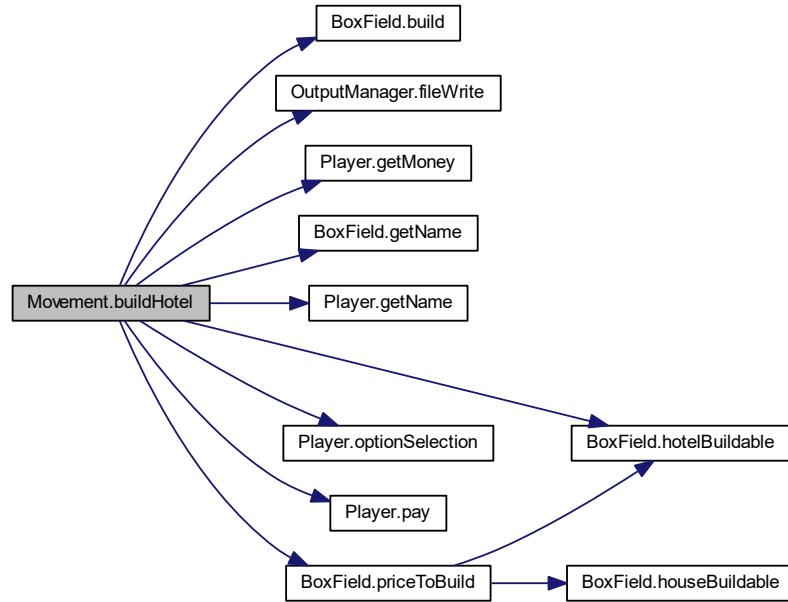
L'acció per contruir un hotel dins d'una propietat ha estat gestionada.

Paràmetres

<i>field</i>	terreny on s'ha de construir apartaments.
--------------	---

Definició a la línia 411 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.5 `buyField()`

```
void Movement.buyField( )
```

Administrerà la compra d'un terreny on s'ha caigut de tauler.

Precondició

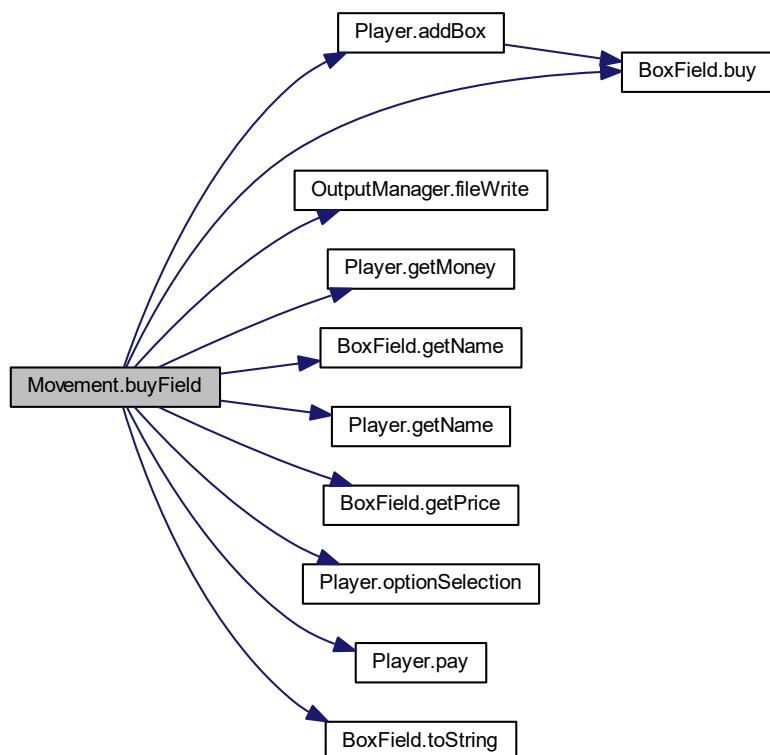
true

Postcondició

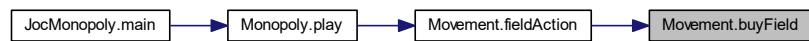
Si el terreny no te propietari i el jugador ho desitja, el terreny serà comprat, altrament passarà.

Definició a la línia 229 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.6 directCommand()

```
void Movement.directCommand( )
```

Administra les accions que s'han de fer si es cau en una Casella de comanda directa.

Precondició

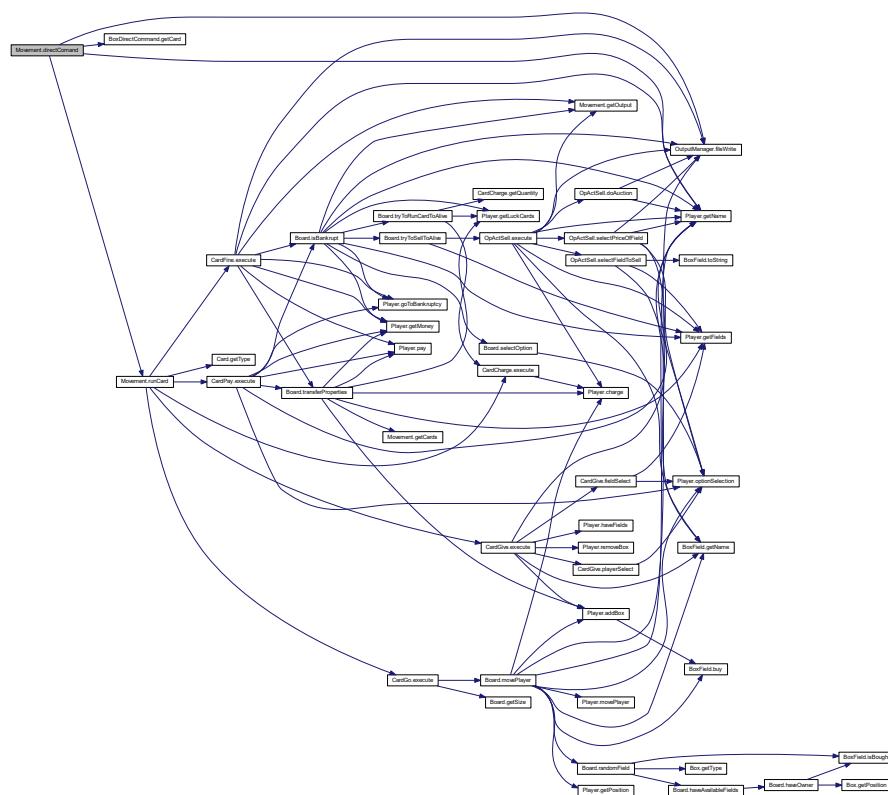
true

Postcondició

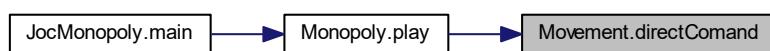
Realitza el moviment dependent del tipus de comanda directa que és.

Definició a la línia 155 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.7 fieldAction()

```
void Movement.fieldAction ( )
```

Administra les accions que s'han de fer si es cau en una Casella de terreny Field.

Precondició

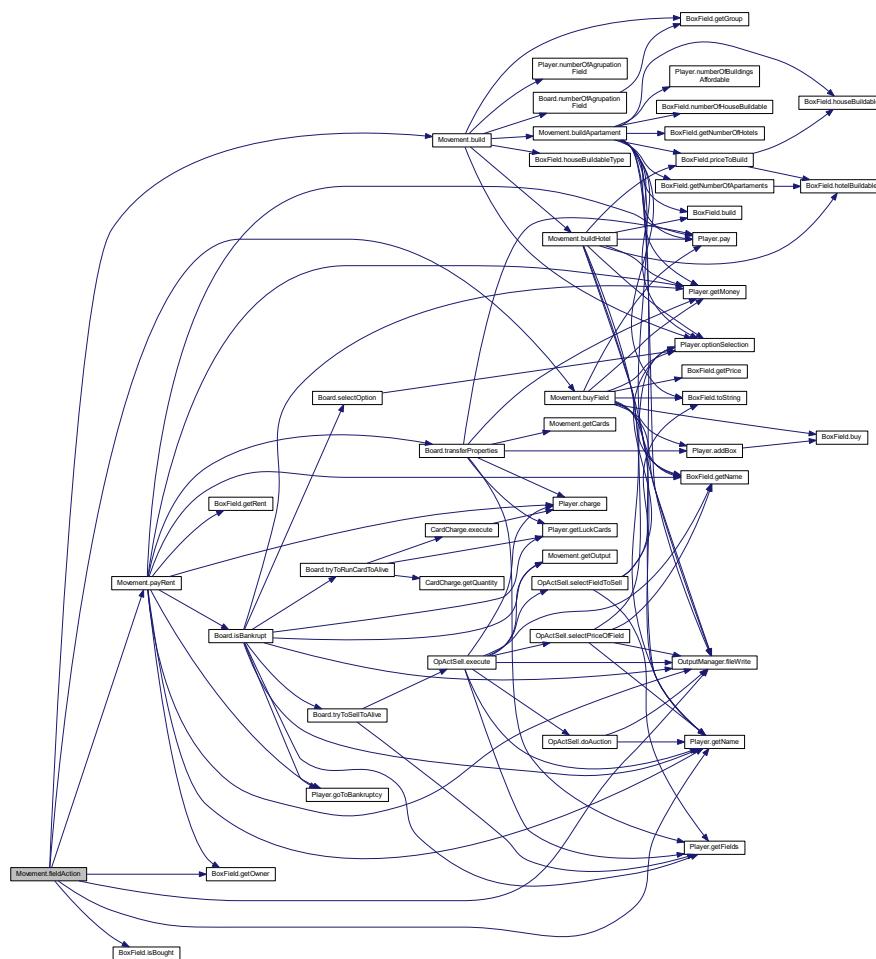
true

Postcondició

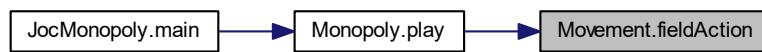
Les accions han sigut realitzades.

Definició a la línia 54 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.8 getCards()

`ArrayList<Card> Movement.getCards()`

Retorna les targetes sort del joc del [Monopoly](#).

Precondició

true.

Postcondició

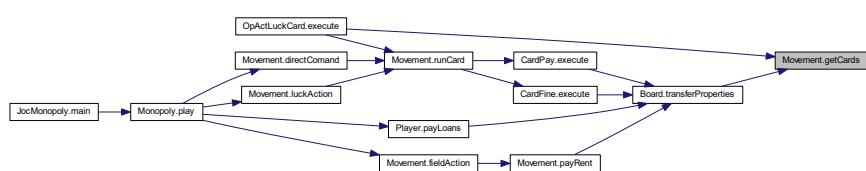
Les targetes sort han estat retornades.

Retorna

`ArrayList` de les targetes sort de [Monopoly](#).

Definició a la línia 452 del fitxer Movement.java.

Gràfic de crides a aquesta funció:



5.19.3.9 getOutput()

`OutputManager Movement.getOutput()`

Retorna la classe de sortida de dades.

Precondició

true.

Postcondició

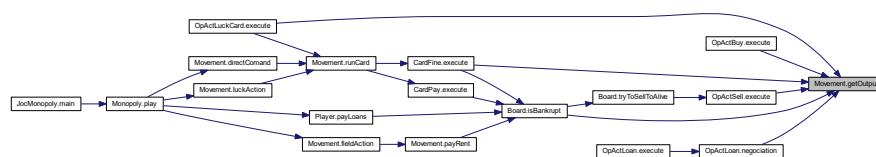
el objecte ha sigut retornat

Retorna

output manager

Definició a la línia 462 del fitxer Movement.java.

Gràfic de crides a aquesta funció:



5.19.3.10 luckAction()

```
void Movement.luckAction ( )
```

Administra les accions que s'han de fer si es cau en una Casella de sort.

Precondició

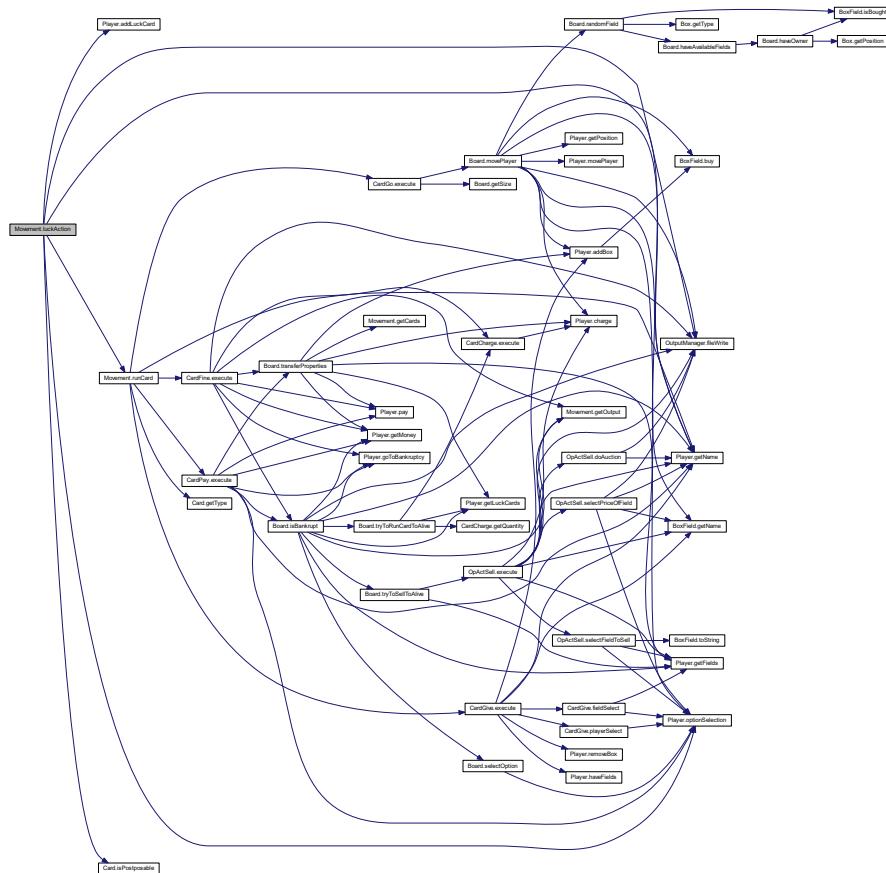
true

Postcondició

S'executarà la targeta de sort agafada de les targetes sort. En cas que la targeta sigui posposable pot guardar-se o utilitzar-se, altrament s'executara.

Definició a la línia 118 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.11 optionalActions()

```
void Movement.optionalActions (
    ArrayList< optionalActions > possible_actions )
```

Administra totes les possibles accions opcionals.

Precondició

```
true
```

Postcondició

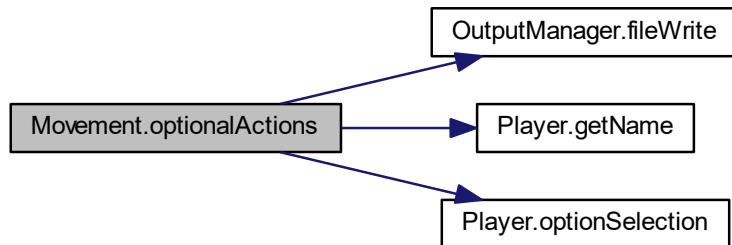
L'acció opcional ha estat executada.

Paràmetres

<i>possible_actions</i>	ArrayList amb totes les accions possibles que el Jugador pot fer.
-------------------------	---

Definició a la línia 205 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.19.3.12 payRent()**

```
void Movement.payRent( )
```

Administrerà el pagament dels lloguers en els terrenys amb propietaris.

Precondició

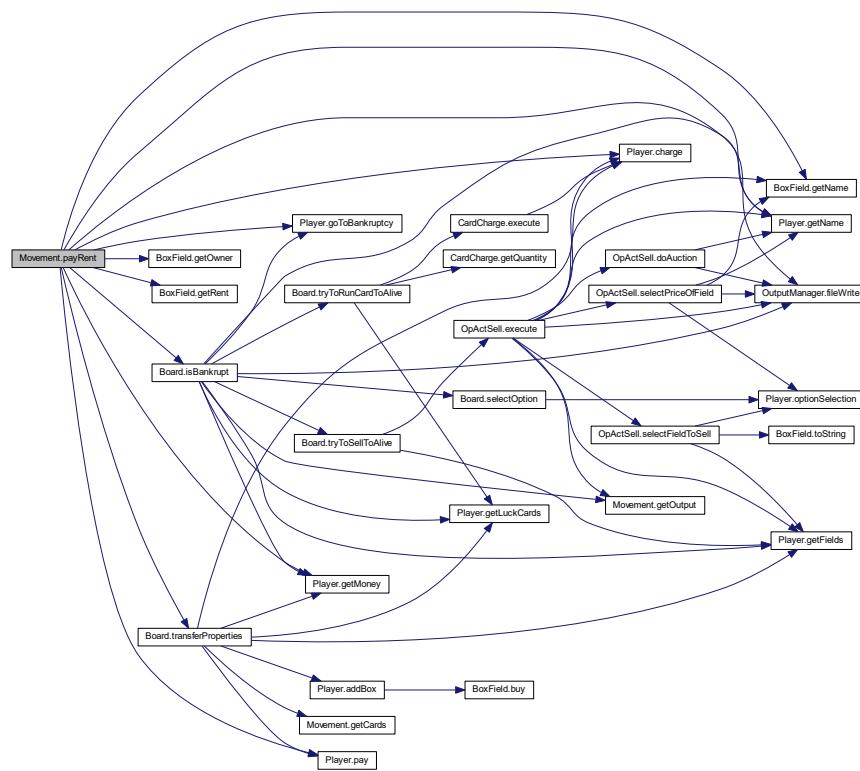
true

Postcondició

El jugador ha pagat el lloguer del terreny on ha caigut al seu Jugador propietari.

Definició a la línia 283 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.13 runCard()

```
void Movement.runCard (
    Card card )
```

Administra les accions que s'han de fer depenent de la targeta que et toqui.

Precondició

card != null

Postcondició

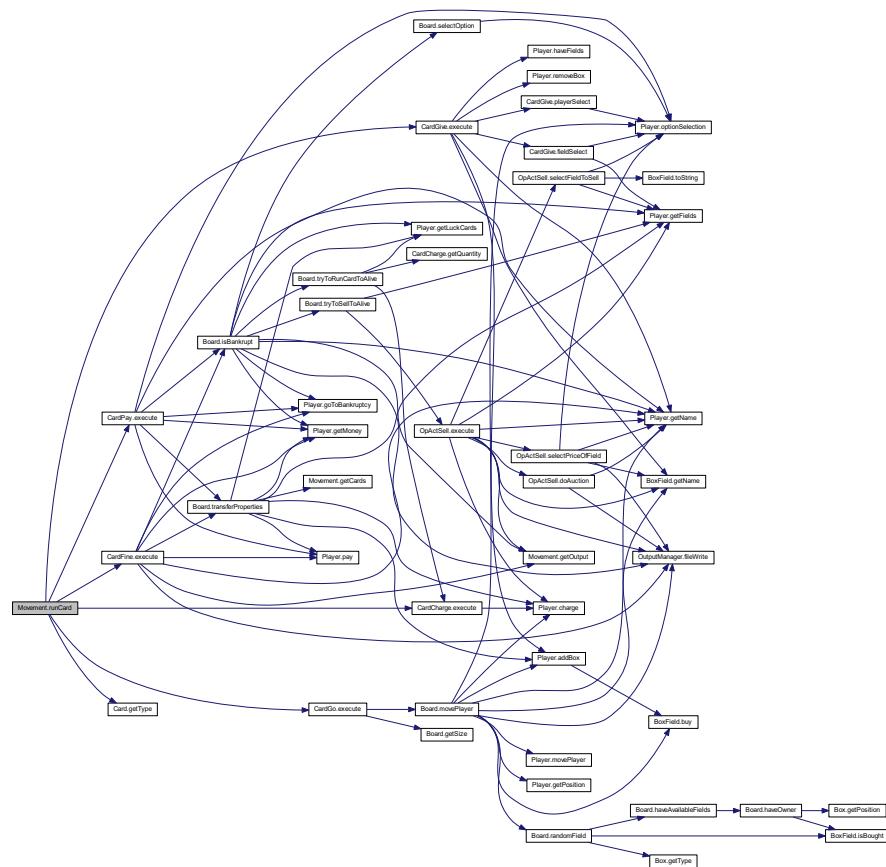
La targeta card ha estat executada.

Paràmetres

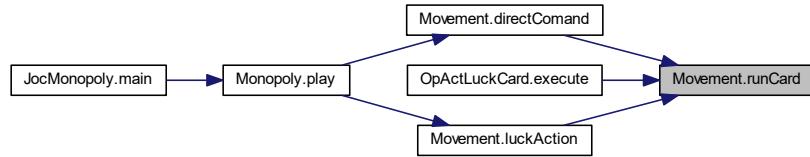
card	targeta que s'ha d'executar.
------	------------------------------

Definició a la línia 169 del fitxer Movement.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.3.14 startAction()

```
void Movement.startAction ( )
```

Informa al Jugador que ha caigut a la casella de sortida Start.

Precondició

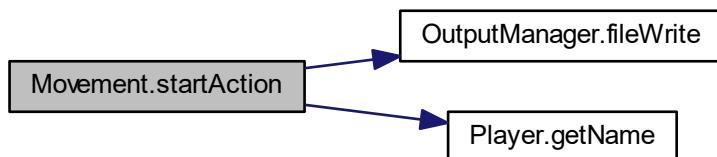
`true`

Postcondició

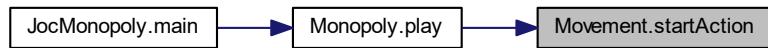
La informació ha sigut mostrada per pantalla.

Definició a la línia 43 del fitxer `Movement.java`.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.19.4 Documentació de les Dades Membre

5.19.4.1 active_player

```
final Player Movement.active_player [private]
```

Jugador que està realitzant el moviment.

Definició a la línia 14 del fitxer Movement.java.

5.19.4.2 cards

```
final ArrayList<Card> Movement.cards [private]
```

Llista de targetes sort de [Monopoly](#).

Definició a la línia 18 del fitxer Movement.java.

5.19.4.3 current_box

```
final Box Movement.current_box [private]
```

Casella actual.

Definició a la línia 13 del fitxer Movement.java.

5.19.4.4 output

```
final OutputManager Movement.output [private]
```

Definició a la línia 19 del fitxer Movement.java.

5.19.4.5 start_rewards

```
final ArrayList<String> Movement.start_rewards [private]
```

Llista de recompenses de la casella Sortida.

Definició a la línia 17 del fitxer Movement.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [Movement.java](#)

5.20 Referència de la Classe OpActBuy

Diagrama d'Herència per a OpActBuy:

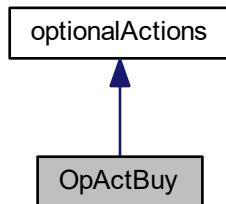
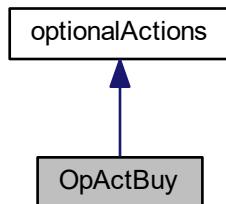


Diagrama de col·laboració per a OpActBuy:



Mètodes públics

- **OpActBuy ()**
Constructor de Buy.
- String **toString ()**
toString per mostrar la descripció de l'acció Buy per text.
- boolean **execute (ArrayList< Player > players, Player current_player, Movement m)**
Mètode per executar el procés de compra un terreny en propietat d'un jugador.

Mètodes Privats

- Player **playerSelect (Player current_player, ArrayList< Player > players, OutputManager output)**
Mostra tots els jugadors amb terrenys i deixa escollir a un d'ells per comprar-li un terreny.
- BoxField **fieldSelect (Player buy_player)**
Mostra tots els terrenys del jugador i deixa escollir a un d'ells per comprar'l-ho.
- int **startOffer (Player current_player, BoxField buy_field, OutputManager output)**

Demana al jugador el valor de la seva oferta inicial.

- boolean `buyNegociation` (`Player` `current_player`, `Player` `buy_player`, `BoxField` `buy_field`, int `initial_offer`, `OutputManager` `output`)

Gestiona la negociació de la compra del terreny.

- boolean `buy` (`Player` `current_player`, `Player` `buy_player`, `BoxField` `buy_field`, int `final_offer`)

Gestiona la negociació de la compra del terreny.

5.20.1 Descripció Detallada

Definició a la línia 10 del fitxer OpActBuy.java.

5.20.2 Documentació del Constructor i el Destructor

5.20.2.1 OpActBuy()

`OpActBuy.OpActBuy ()`

Constructor de Buy.

Precondició

`true`

Postcondició

Crea una acció opcional Buy.

Definició a la línia 17 del fitxer OpActBuy.java.

5.20.3 Documentació de les Funcions Membre

5.20.3.1 buy()

```
boolean OpActBuy.buy (
    Player current_player,
    Player buy_player,
    BoxField buy_field,
    int final_offer ) [private]
```

Gestiona la negociació de la compra del terreny.

Paràmetres

<i>current_player</i>	Jugador que fa la compra
<i>buy_player</i>	Jugador que ven el terreny
<i>buy_field</i>	Terreny a comprar
<i>final_offer</i>	Oferta final

Retorna

true si s'ha comprat el terreny, false altrament

Precondició

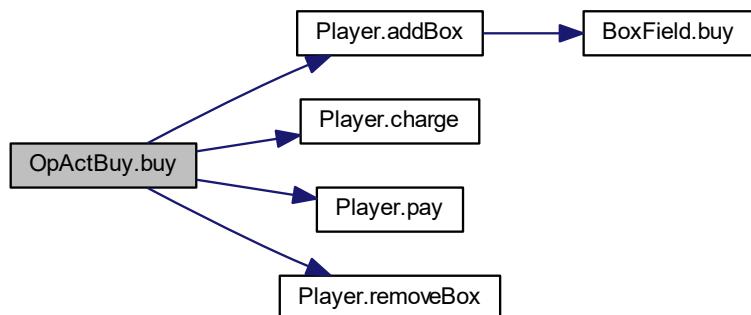
true

Postcondició

el boolean indicant si la compra s'ha dut a terme s'ha retornat

Definició a la línia 182 del fitxer OpActBuy.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.20.3.2 buyNegociation()

```
boolean OpActBuy.buyNegociation (
    Player current_player,
    Player buy_player,
    BoxField buy_field,
    int initial_offer,
    OutputManager output ) [private]
```

Gestiona la negociació de la compra del terreny.

Paràmetres

<i>current_player</i>	Jugador que fa la compra
<i>buy_player</i>	Jugador que ven el terreny
<i>buy_field</i>	Terreny a comprar
<i>initial_offer</i>	Oferta inicial
<i>output</i>	sortida al fitxer

Retorna

true si s'ha comprat el terreny, false altrament

Precondició

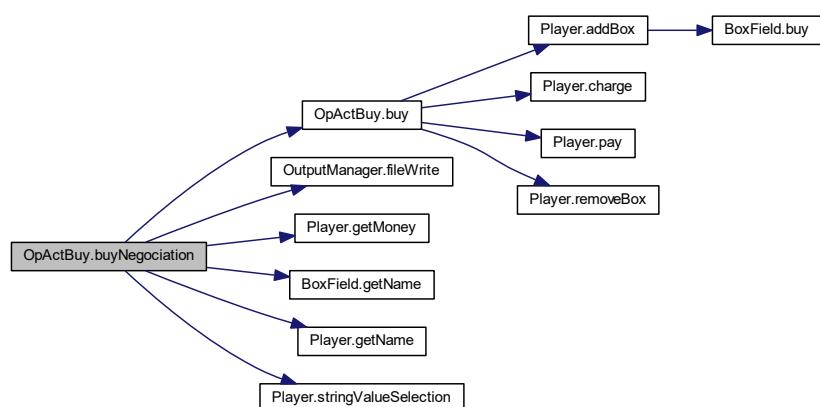
true

Postcondició

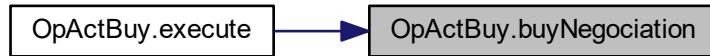
el boolean indicant si la compra s'ha dut a terme s'ha retornat

Definició a la línia 145 del fitxer OpActBuy.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.20.3.3 execute()

```
boolean OpActBuy.execute (
    ArrayList< Player > players,
    Player current_player,
    Movement m )
```

Mètode per executar el procés de compra un terreny en propietat d'un jugador.

Paràmetres

<i>players</i>	ArrayList de jugadors que estan jugant al Monopoly.
<i>current_player</i>	Jugador que fa la compra.
<i>m</i>	Moviment que crida Buy, en aquesta implementació, <i>m</i> no és usada però s'ha de passar.

Retorna

true si el procés s'ha realitzat, false altrament.

Precondició

Jugador *current_player* != null i Moviment *m* != null

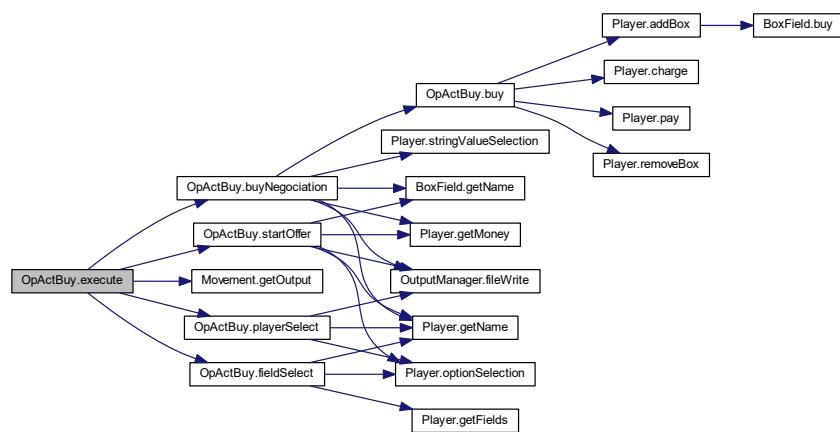
Postcondició

Una propietat de algun jugador ha estat comprada o la acció s'ha cancelat

Reimplementat de [optionalActions](#).

Definició a la línia 39 del fitxer OpActBuy.java.

Gràfic de crides d'aquesta funció:



5.20.3.4 fieldSelect()

```
BoxField OpActBuy.fieldSelect (
    Player buy_player ) [private]
```

Mostra tots els terreys del jugador i deixa escollir a un d'ells per comprar'l-ho.

Paràmetres

<code>buy_player</code>	Jugador que ven el terreny
-------------------------	----------------------------

Retorna

el terreny seleccionat per el jugador o null si no es pot

Precondició

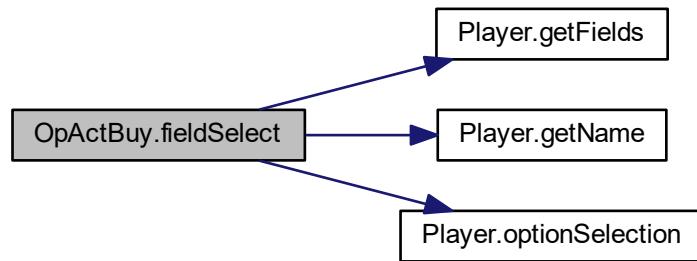
true

Postcondició

El terreny seleccionat ha estat retornat, o en la seva absència, s'ha retornat un terreny null

Definició a la línia 91 del fitxer OpActBuy.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.20.3.5 `playerSelect()`

```

Player OpActBuy.playerSelect (
    Player current_player,
    ArrayList< Player > players,
    OutputManager output ) [private]
  
```

Mostra tots els jugadors amb terrenys i deixa escollir a un d'ells per comprar-li un terreny.

Paràmetres

<code>current_player</code>	Jugador que fa la compra.
<code>players</code>	ArrayList de jugadors que estan jugant al Monopoly.
<code>output</code>	sortida al fitxer

Retorna

el jugador seleccionat o null si no n'hi ha cap

Precondició

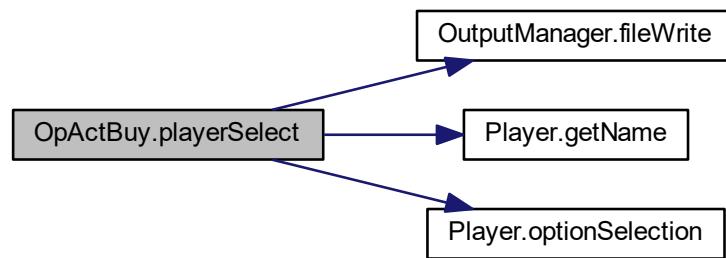
```
true
```

Postcondició

El jugador seleccionat ha estat retornat, o en la seva absència, s'ha retornat un jugador null

Definició a la línia 61 del fitxer OpActBuy.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.20.3.6 startOffer()**

```
int OpActBuy.startOffer (
    Player current_player,
    BoxField buy_field,
    OutputManager output )  [private]
```

Demana al jugador el valor de la seva oferta inicial.

Paràmetres

<i>current_player</i>	Jugador que fa la compra.
<i>buy_field</i>	Terreny a comprar
<i>output</i>	sortida al fitxer

Retorna

el valor de la oferta inicial o un -1 en cas de error (no tractat ja que no arriba mai a retornar-lo)

Precondició

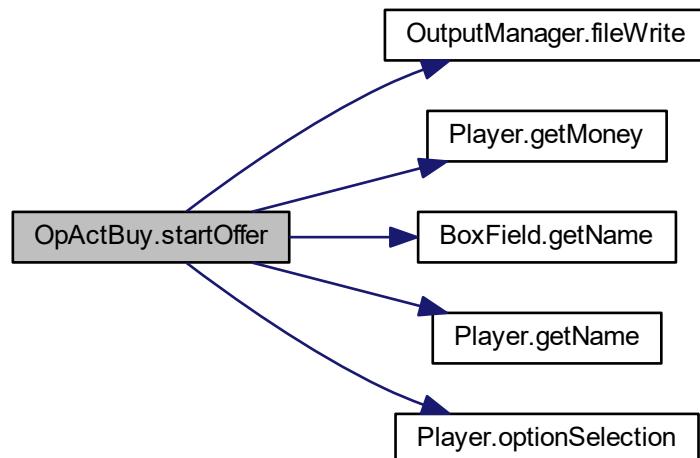
true

Postcondició

El valor de la oferta ha estat retornat i el valor > 0 i $<$ la quantitat de diners del jugador comprador

Definició a la línia 117 del fitxer OpActBuy.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.20.3.7 `toString()`

`String OpActBuy.toString ()`

`toString` per mostrar la descripció de l'acció Buy per text.

Retorna

String de la sortida per pantalla de Buy

Precondició

`true`

Postcondició

Mostra la descripció de Buy

Reimplementat de [optionalActions](#).

Definició a la línia 26 del fitxer OpActBuy.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [OpActBuy.java](#)

5.21 Referència de la Classe OpActLoan

Diagrama d'Herència per a OpActLoan:

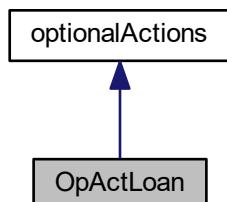
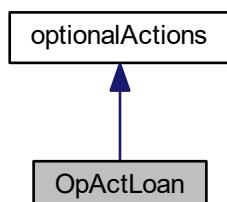


Diagrama de col·laboració per a OpActLoan:



Mètodes públics

- **OpActLoan ()**
Constructor per defecte de Loan.
- **String toString ()**
toString per mostrar la descripció de l'acció LuckCard per text.
- **boolean execute (ArrayList< Player > players, Player current_player, Movement aux)**
Mètode per executar el procés de demanar un préstec a un Jugador.

Mètodes Privats

- **Player playerSelect (ArrayList< Player > players, Player current_player)**
Mostra tots els jugadors i deixa escollir a un d'ells.
- **int offer (Player current_player, Player loan_player)**
Demana al jugador la quantitat que vol demanar.
- **boolean negociation (Player current_player, Player loan_player, int current_offer, Movement aux)**
Gestiona la negociació de els torns i interessos del prestec.
- **boolean giveLoan (Player loan_player, Player current_player, int current_offer, int interests, int turns)**
Afageix un prestec al jugador que el reb i treu els diners al jugador que el dona.

5.21.1 Descripció Detallada

Definició a la línia 9 del fitxer OpActLoan.java.

5.21.2 Documentació del Constructor i el Destructor

5.21.2.1 OpActLoan()

OpActLoan.OpActLoan ()

Constructor per defecte de Loan.

Precondició

true

Postcondició

Crea una acció opcional Loan.

Definició a la línia 16 del fitxer OpActLoan.java.

5.21.3 Documentació de les Funcions Membre

5.21.3.1 execute()

```
boolean OpActLoan.execute (
    ArrayList< Player > players,
    Player current_player,
    Movement aux )
```

Mètode per executar el procés de demanar un préstec a un Jugador.

Precondició

`current_player != null`

Postcondició

S'ha demanat un préstec a un Jugador.

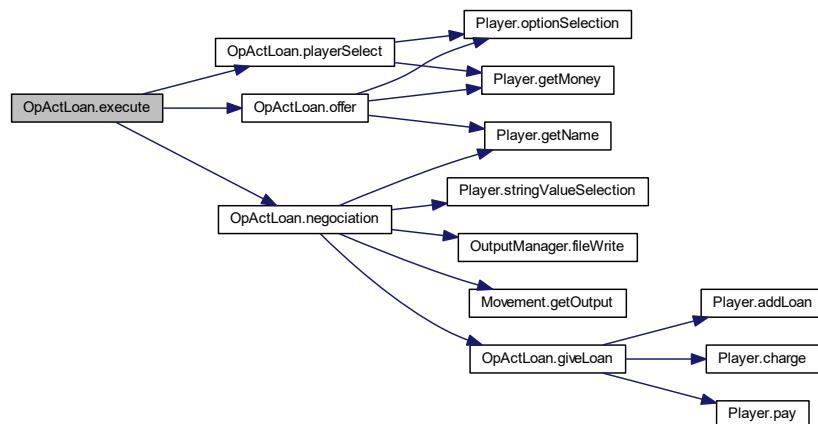
Retorna

`true` quan ha acabat de demanar el prestec.

Reimplementat de [optionalActions](#).

Definició a la línia 35 del fitxer OpActLoan.java.

Gràfic de crides d'aquesta funció:



5.21.3.2 giveLoan()

```
boolean OpActLoan.giveLoan (
    Player loan_player,
    Player current_player,
    int current_offer,
    int interests,
    int turns ) [private]
```

Afageix un préstec al jugador que el reb i treu els diners al jugador que el dona.

Paràmetres

<i>current_player</i>	Jugador que reb el prestec
<i>loan_player</i>	Jugador que dona el prestec
<i>current_offer</i>	Valor del prestec
<i>interests</i>	Tant per cent de interes del prestes
<i>turns</i>	Torns a retornar el prestec

Retorna

true si s'ha realitzat correctament el prestec

Precondició

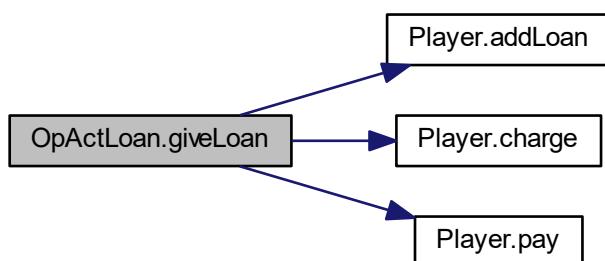
true

Postcondició

el prestec s'ha realitzat

Definició a la línia 129 del fitxer OpActLoan.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.21.3.3 negociação()

```
boolean OpActLoan.negociação (
    Player current_player,
    Player loan_player,
    int current_offer,
    Movement aux ) [private]
```

Gestiona la negociació de els torns i interessos del prestatge.

Paràmetres

<i>current_player</i>	Jugador que reb el prestatge
<i>loan_player</i>	Jugador que dona el prestatge
<i>current_offer</i>	Valor que es demana del prestatge

Retorna

true si \loan_player ha acceptat o false altrament

Precondició

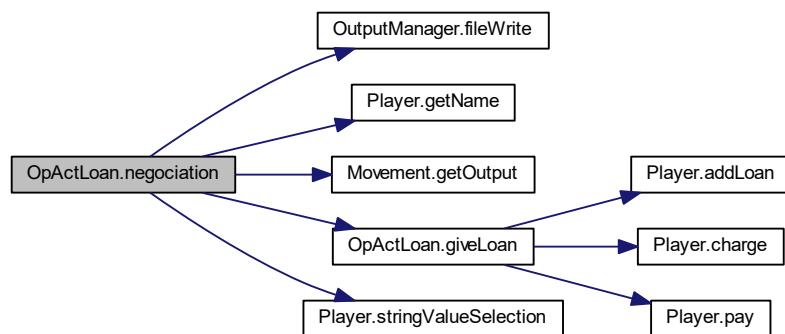
true

Postcondició

la negociació de els torns i interessos del prestatge s'ha realitzat

Definició a la línia 91 del fitxer OpActLoan.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.21.3.4 offer()

```
int OpActLoan.offer (
    Player current_player,
    Player loan_player ) [private]
```

Demana al jugador la quantitat que vol demanar.

Paràmetres

<i>current_player</i>	Jugador que reb el prestatge
<i>loan_player</i>	Jugador que dona el prestatge

Retorna

valor inicial del prestatge

Precondició

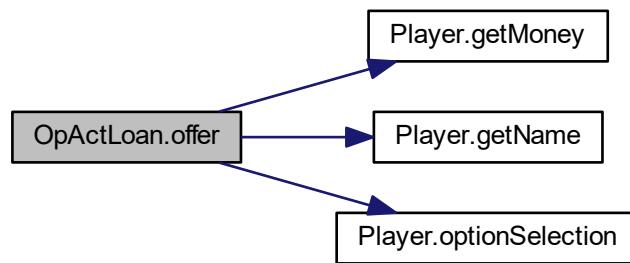
true

Postcondició

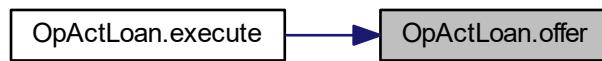
el valor ha estat retornat

Definició a la línia 70 del fitxer OpActLoan.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.21.3.5 `playerSelect()`

```

Player OpActLoan.playerSelect (
    ArrayList<Player> players,
    Player current_player ) [private]
  
```

Mostra tots els jugadors i deixa escollir a un d'ells.

Paràmetres

<code>players</code>	ArrayList de jugadors que estan jugant al Monopoly .
<code>current_player</code>	Jugador que reb el prestatge

Retorna

el jugador seleccionat o null si no n'hi ha cap

Precondició

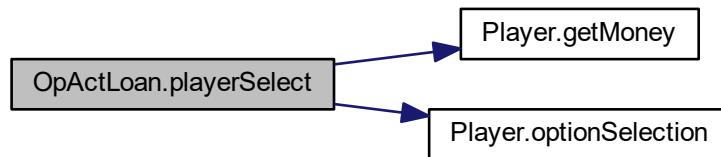
`players size > 1 i algun de ells != jugador actual`

Postcondició

El jugador seleccionat ha estat retornat (mai retorna un jugador null per la precondició)

Definició a la línia 51 del fitxer OpActLoan.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

**5.21.3.6 `toString()`**

```
String OpActLoan.toString ( )
```

`toString` per mostrar la descripció de l'acció LuckCard per text.

Precondició

`true`

Postcondició

Mostra la descripció de LuckCard.

Retorna

String de la sortida per pantalla de LuckCard.

Reimplementat de [optionalActions](#).

Definició a la línia 25 del fitxer OpActLoan.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [OpActLoan.java](#)

5.22 Referència de la Classe OpActLuckCard

Diagrama d'Herència per a OpActLuckCard:

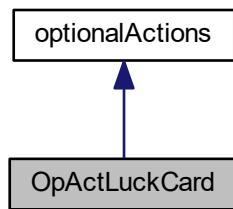
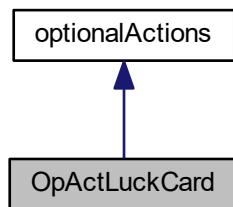


Diagrama de col·laboració per a OpActLuckCard:



Mètodes públics

- [OpActLuckCard \(\)](#)
Constructor per defecte de LuckCard.
- [String toString \(\)](#)
toString per mostrar la descripció de l'acció LuckCard per text.
- [boolean execute \(ArrayList< Player > players, Player current_player, Movement aux\)](#)
Mètode per executar el procés d'utilitzar una targeta sort en la seva propietat al final del torn.

5.22.1 Descripció Detallada

Definició a la línia 12 del fitxer OpActLuckCard.java.

5.22.2 Documentació del Constructor i el Destructor

5.22.2.1 OpActLuckCard()

```
OpActLuckCard.OpActLuckCard ( )
```

Constructor per defecte de LuckCard.

Precondició

```
true
```

Postcondició

Crea una acció opcional LuckCard

Definició a la línia 19 del fitxer OpActLuckCard.java.

5.22.3 Documentació de les Funcions Membre

5.22.3.1 execute()

```
boolean OpActLuckCard.execute (  
    ArrayList< Player > players,  
    Player current_player,  
    Movement aux )
```

Mètode per executar el procés d'utilitzar una targeta sort en la seva propietat al final del torn.

Precondició

```
true
```

Postcondició

Crea [Monopoly](#) amb els atributs entrats.

Paràmetres

<i>players</i>	Llista de Jugadors jugant al Monopoly.
<i>current_player</i>	Jugador actual del torn.
<i>aux</i>	moviment que li passem ja que en aquest cas sera el mateix moviment que executara una carta.

Retorna

true quan acabi d'executar la carta de sort.

Reimplementat de [optionalActions](#).

Definició a la línia 41 del fitxer OpActLuckCard.java.

Gràfic de crides d'aquesta funció:



5.22.3.2 `toString()`

```
String OpActLuckCard.toString ( )
```

`toString` per mostrar la descripció de l'acció LuckCard per text.

Precondició

true

Postcondició

Mostra la descripció de LuckCard.

Retorna

String de la sortida per pantalla de LuckCard.

Reimplementat de [optionalActions](#).

Definició a la línia 28 del fitxer OpActLuckCard.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [OpActLuckCard.java](#)

5.23 Referència de la Classe OpActSell

Diagrama d'Herència per a OpActSell:

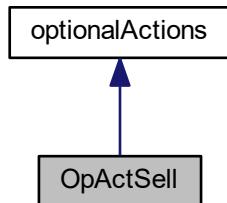
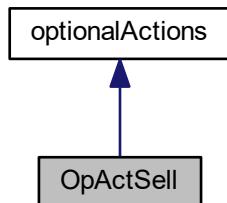


Diagrama de col·laboració per a OpActSell:



Mètodes públics

- **OpActSell ()**
Constructor de Sell.
- String **toString ()**
toString per mostrar la descripció de l'acció Sell per text.
- boolean **execute (ArrayList< Player > players, Player current_player, Movement m)**
Mètode per executar el procés de vendre un terreny en propietat d'un jugador.

Mètodes Privats

- Pair< Player, Integer > **doAuction (ArrayList< Player > players, Player current_player, BoxField field_to_sell, OutputManager output, int sell_price)**
Mètode per executar la subhasta d'una venta.
- int **selectPriceOfField (Player current_player, BoxField field_to_sell, OutputManager output)**
Mètode per seleccionar el preu del terreny a vendre.
- **BoxField selectFieldToSell (Player current_player)**
Mètode per seleccionar el terreny que es vol vendre.

5.23.1 Descripció Detallada

Definició a la línia 13 del fitxer OpActSell.java.

5.23.2 Documentació del Constructor i el Destructor

5.23.2.1 OpActSell()

OpActSell.OpActSell ()

Constructor de Sell.

Precondició

true

Postcondició

Crea una acció opcional Sell.

Definició a la línia 20 del fitxer OpActSell.java.

5.23.3 Documentació de les Funcions Membre

5.23.3.1 doAuction()

```
Pair<Player, Integer> OpActSell.doAuction (
    ArrayList< Player > players,
    Player current_player,
    BoxField field_to_sell,
    OutputManager output,
    int sell_price ) [private]
```

Mètode per executar la subhasta d'una venda.

Precondició

true

Postcondició

La subhasta ha estat realitzada.

Paràmetres

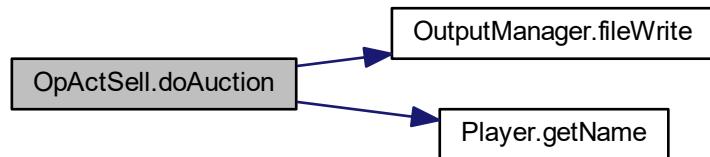
<i>players</i>	ArrayList de jugadors que estan jugant al Monopoly.
<i>current_player</i>	Jugador que fa la venda.
<i>field_to_sell</i>	Terreny a subastar.
<i>output</i>	Fitxer desenvolupament partida.
<i>sell_price</i>	Preu inicial de venta.

Retorna

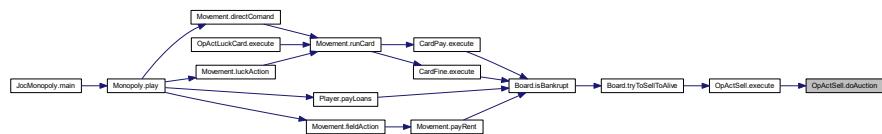
Retorna el jugador guanyador i el preu pel que l'ha comprat, altrament torna jugador null.

Definició a la línia 87 del fitxer OpActSell.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.23.3.2 execute()

```
boolean OpActSell.execute (
    ArrayList< Player > players,
    Player current_player,
    Movement m )
```

Mètode per executar el procés de vendre un terreny en propietat d'un jugador.

Precondició

Jugador current_player != null i Moviment m != null

Postcondició

Una propietat del jugador ha estat venuda al guanyador de la subasta, en cas contrari no ha estat venuda i la subasta no té guanyador.

Paràmetres

<i>players</i>	ArrayList de jugadors que estan jugant al Monopoly.
<i>current_player</i>	Jugador que fa la venda.
<i>m</i>	Moviment que crida Sell, en aquesta implementació, <i>m</i> no és usada però s'ha de passar.

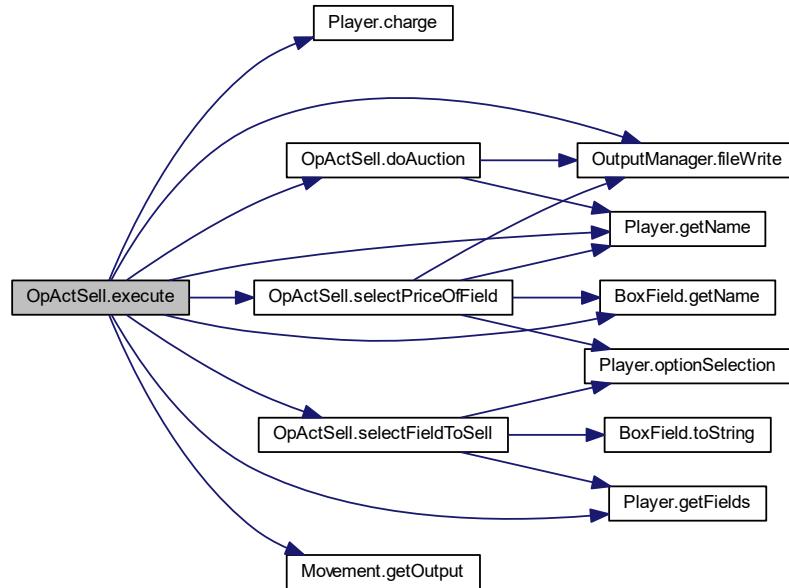
Retorna

true si el procés s'ha realitzat, false altrement.

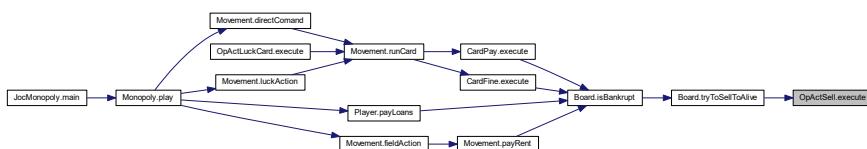
Reimplementat de [optionalActions](#).

Definició a la línia 43 del fitxer OpActSell.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.23.3.3 selectFieldToSell()

```
BoxField OpActSell.selectFieldToSell (
    Player current_player ) [private]
```

Mètode per sel·leccionar el terreny que es vol vendre.

Precondició

```
current_player != null
```

Postcondició

Propietat a vendre sel·leccionada.

Paràmetres

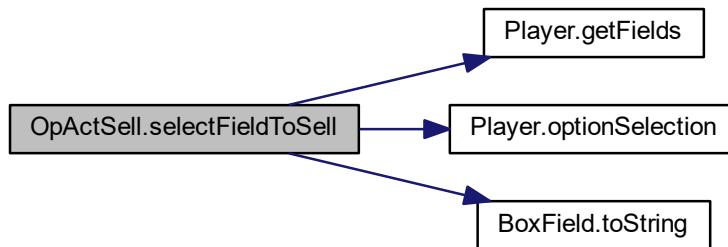
<i>current_player</i>	Jugador que fa la venda.
-----------------------	--------------------------

Retorna

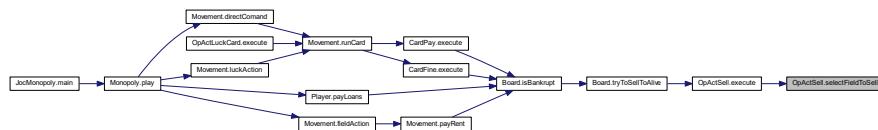
Retorna un terreny *Field*, null en cas de cancel·lar-se.

Definició a la línia 155 del fitxer OpActSell.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:

5.23.3.4 `selectPriceOfField()`

```

int OpActSell.selectPriceOfField (
    Player current_player,
    BoxField field_to_sell,
    OutputManager output ) [private]
  
```

Mètode per seleccionar el preu del terreny a vendre.

Precondició

`current_player != null`

Postcondició

Preu a vendre seleccionat.

Paràmetres

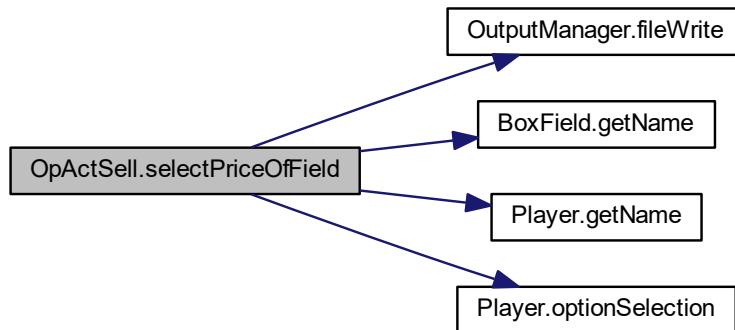
<i>current_player</i>	Jugador que fa la venda.
<i>field_to_sell</i>	Terreny a vendre
<i>output</i>	Fitxer a escriure

Retorna

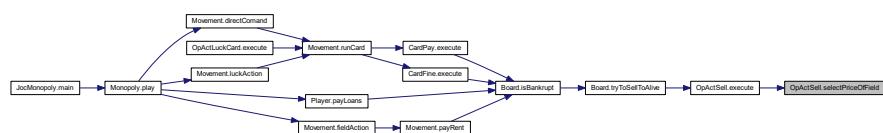
Retorna el preu *sell_price* a vendre.

Definició a la línia 139 del fitxer OpActSell.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.23.3.5 `toString()`

```
String OpActSell.toString ( )
```

`toString` per mostrar la descripció de l'acció Sell per text.

Precondició

true

Postcondició

Mostra la descripció de Sell

Retorna

String de la sortida per pantalla de Sell

Reimplementat de [optionalActions](#).

Definició a la línia 29 del fitxer OpActSell.java.

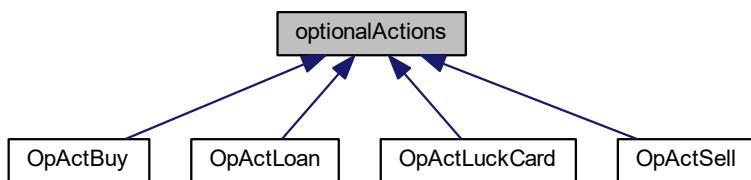
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [OpActSell.java](#)

5.24 Referència de la Classe optionalActions

Interfície general d'accions opcionals del joc de [Monopoly](#).

Diagrama d'Herència per a optionalActions:



Mètodes públics

- String [toString \(\)](#)
toString que tindrà cada acció opcional per tal de mostrar la seva funció.
- boolean [execute \(ArrayList< Player > players, Player current_player, Movement aux\)](#)
Execució d'una acció opcional, dependent de quina, farà una funció o altre descrita en el seu fitxer java corresponent.

5.24.1 Descripció Detallada

Interfície general d'accions opcionals del joc de [Monopoly](#).

Definició a la línia 8 del fitxer optionalActions.java.

5.24.2 Documentació de les Funcions Membre

5.24.2.1 execute()

```
boolean optionalActions.execute (
    ArrayList< Player > players,
    Player current_player,
    Movement aux )
```

Execució d'una acció opcional, depenent de quina, farà una funció o altre descrita en el seu fitxer java corresponent.

Precondició

true

Postcondició

Retorna true si l'acció s'ha realitzat correctament, false altrament.

Retorna

true si s'ha pogut realitzar l'acció, false altrament.

Reimplementat a [OpActSell](#), [OpActBuy](#), [OpActLuckCard](#) i [OpActLoan](#).

5.24.2.2 toString()

```
String optionalActions.toString ( )
```

toString que tindrà cada acció opcional per tal de mostrar la seva funció.

Precondició

true

Postcondició

La informació ha sigut mostrada per pantalla.

Retorna

Informació específica de cada acció opcional.

Reimplementat a [OpActSell](#), [OpActLuckCard](#), [OpActLoan](#) i [OpActBuy](#).

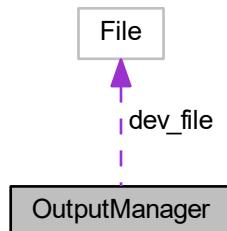
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [optionalActions.java](#)

5.25 Referència de la Classe OutputManager

Classe que s'encarrega de gestionar les sortides en el nostre cas al fitxer de desenvolupament de partida.

Diagrama de col·laboració per a OutputManager:



Mètodes públics

- `OutputManager ()`

Constructor `OutputManager`. Crea un directori `saves` si no està creat i crea un fitxer de desenvolupament de partida amb nom. `logs_x` on `x` és un número correlatiu.

- void `fileWrite (String line)`

Escriu text `line` al fitxer de desenvolupament de partida `dev_file`.

Atributs Privats

- File `dev_file`

Fitxer desenvolupament de la partida.

5.25.1 Descripció Detallada

Classe que s'encarrega de gestionar les sortides en el nostre cas al fitxer de desenvolupament de partida.

Definició a la línia 11 del fitxer OutputManager.java.

5.25.2 Documentació del Constructor i el Destructor

5.25.2.1 OutputManager()

```
OutputManager.OutputManager ( )
```

Constructor [OutputManager](#). Crea un directori `saves` si no esta creat i crea un fitxer de desenvolupament de partida amb nom. `logs_x` on `x` es un numero correlatiu.

Precondició

`true`

Postcondició

El fitxer de logs ha estat creat.

Definició a la línia 21 del fitxer OutputManager.java.

5.25.3 Documentació de les Funcions Membre

5.25.3.1 fileWrite()

```
void OutputManager.writeFile (
    String line )
```

Escriu text `line` al fitxer de desenvolupament de partida `dev_file`.

Precondició

`true`

Postcondició

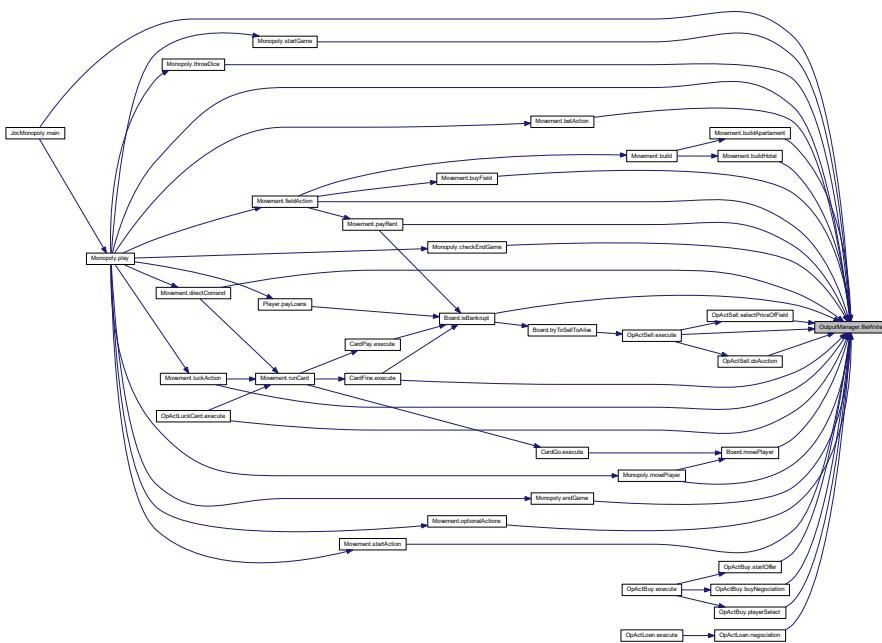
la linea de text `line` ha estat escrita a `dev_file`.

Paràmetres

<code>line</code>	text que s'ha d'escriure al fitxer.
-------------------	-------------------------------------

Definició a la línia 44 del fitxer OutputManager.java.

Gràfic de crides a aquesta funció:



5.25.4 Documentació de les Dades Membre

5.25.4.1 dev_file

File OutputManager.dev_file [private]

Fitxer desenvolupament de la partida.

Definició a la línia 13 del fitxer OutputManager.java.

La documentació d'aquesta classe es va generar a partir del seqüent fitxer:

- OutputManager.java

5.26 Referència de la Classe Player

Classe de Jugador del [Monopoly](#). En aquesta classe hi haurà tota la informació del jugador i algunes accions que pot fer el jugador en si.

Diagrama d'Herència per a Player:

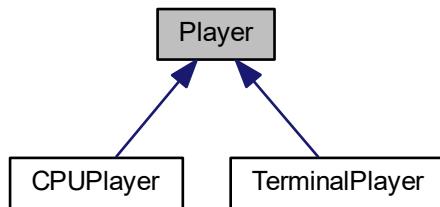
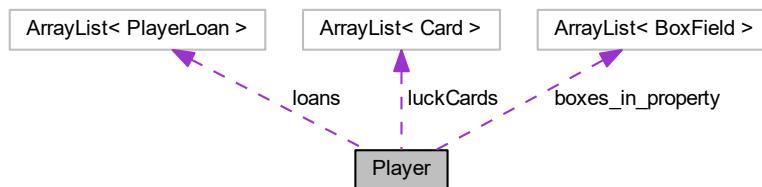


Diagrama de col·laboració per a Player:



Mètodes públics

- **Player** (String `name`, int `initial_money`, int `initial_position`, String `type`)

Constructor de Player.
- void **pay** (int `amount`)

El Jugador paga una quantitat de diners amount determinada.
- void **charge** (int `amount`)

El jugador cobra una quantitat de diners amount determinada.
- List< Card > **getLuckCards** ()

Retorna en una llista les targetes sort luckCards del Jugador.
- void **addLuckCard** (Card `card`)

Afegeix una targeta sort card a la propietat del Jugador.
- void **removeLuckCard** (Card `card`)

Elimina una targeta sort card de la propietat del Jugador.
- String **getName** ()

Retorna el nom del Jugador el qual la crida.
- String **getType** ()

Retorna el tipus de Jugador el qual la crida.
- int **getPosition** ()

Retorna la posició del Jugador el qual la crida.
- int **getMoney** ()

Retorna la quantitat de diners que té el Jugador.

- **int `numberOfBuildingsAffordable` (int price)**

Retorna els diners que té el Jugador en possessió.
- **void `movePlayer` (int position)**

Calcula el nombre de apartaments que es pot permetre costruir al preu entrat.
- **void `addBox` (BoxField box)**

Mou el Jugador a una posició position final en el tauler.
- **void `removeBox` (Box box)**

Afegeix un terreny a la propietat d'un Jugador.
- **ArrayList< BoxField > `getFields` ()**

Elimina un terreny a la propietat d'un Jugador.
- **ArrayList< BoxField > `getFields` ()**

Retorna la llista boxes_in_property de terrenys que te en propietat el Jugador.
- **void `goToBankruptcy` ()**

Declara el Jugador en fallida.
- **boolean `haveFields` ()**

Retorna si el Jugador té terrenys en propietat o no.
- **ArrayList< PlayerLoan > `getLoans` ()**
- **Boolean `getBankruptcy` ()**

Retorna si el Jugador es en fallida o no.
- **void `addLoan` (Player loaner, int value, int interest, int turns)**

Afegeix un préstec a la llista de préstecs del Jugador.
- **void `payLoans` (Board board, Movement movement)**

Paga els préstecs que té pendents els Jugadors i gestiona els torns dels que encara no s'han de pagar.
- **int `numberOfAgrupationField` (String group_name)**

calcula el numero de terrenys de la agrupacio passada que te en propietat el jugador
- **String `toString` ()**

toString per mostrar l'informació de Player per text.
- **abstract int `optionSelection` (String type, Player player, BoxField field, ArrayList< Integer > options, ArrayList< Player > players, Card card, int value, ArrayList< optionalActions > optional_actions)**
- **abstract String `stringValueSelection` (String type, Player player, BoxField field, int value, int second_value)**

Atributs Privats

- **final String name**

Nom del Jugador.
- **int money**

Diners del Jugador.
- **final ArrayList< BoxField > boxes_in_property = new ArrayList<>()**

Llista terrenys en propietat del Jugador.
- **int position**

Posició del Jugador.
- **ArrayList< Card > luckCards = new ArrayList<>()**

Llista targetes sort.
- **boolean bankrupt = false**

Estat fallida jugador.
- **ArrayList< PlayerLoan > loans = new ArrayList<>()**

Llista de préstecs del Jugador.
- **final String type**

Tipus de jugador (CPU, USER)

5.26.1 Descripció Detallada

Classe de Jugador del [Monopoly](#). En aquesta classe hi haurà tota la informació del jugador i algunes accions que pot fer el jugador en si.

Definició a la línia 10 del fitxer Player.java.

5.26.2 Documentació del Constructor i el Destructor

5.26.2.1 Player()

```
Player.Player (
    String name,
    int initial_money,
    int initial_position,
    String type )
```

Constructor de [Player](#).

Precondició

true

Postcondició

Crea un jugador amb els atributs entrats.

Paràmetres

<i>name</i>	nom del Jugador.
<i>initial_money</i>	quantitat de diners inicials dels que disposa un Jugador.
<i>initial_position</i>	posició inicial del Jugador.

Definició a la línia 30 del fitxer Player.java.

5.26.3 Documentació de les Funcions Membre

5.26.3.1 addBox()

```
void Player.addBox (
    BoxField box )
```

Afegeix un terreny a la propietat d'un Jugador.

Precondició

box != null

Postcondició

box ha estat afegit a la llista de propietats boxes_in_property de Jugador.

Paràmetres

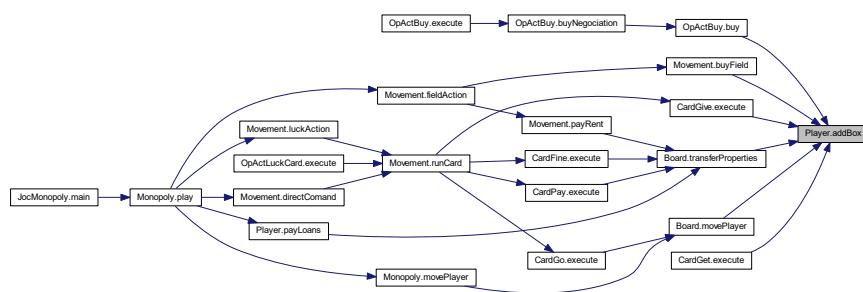
box terreny que serà afegit a la propietat de Jugador.

Definició a la línia 158 del fitxer Player.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.26.3.2 addLoan()

```
void Player.addLoan ( Player loaner,  
                      int value,  
                      int interest,  
                      int turns )
```

Afegeix un préstec a la llista de préstecs del Jugador.

Precondició

```
loaner != null, torns > 0
```

Postcondició

El préstec ha estat afegit a la llista de préstecs `loans` de Jugador.

Paràmetres

<code>loaner</code>	Jugador que ha prestat diners a Jugador.
<code>value</code>	valor a retornar a <code>loaner</code> .
<code>interest</code>	interes del préstec.
<code>torns</code>	numero de torns el qual s'ha de tornar el préstec del Jugador.

Definició a la línia 228 del fitxer Player.java.

Gràfic de crides a aquesta funció:

**5.26.3.3 addLuckCard()**

```
void Player.addLuckCard (
    Card card )
```

Afegeix una targeta sort `card` a la propietat del Jugador.

Precondició

```
card != null
```

Postcondició

La targeta `card` ha estat afegida a la llista de targetes sort del Jugador.

Paràmetres

<code>card</code>	targeta que es vol afegir a les targetes de sort <code>luckCards</code> del Jugador.
-------------------	--

Definició a la línia 73 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.4 charge()

```
void Player.charge (
    int amount )
```

El jugador cobra una quantitat de diners `amount` determinada.

Precondició

`amount >= 0`

Postcondició

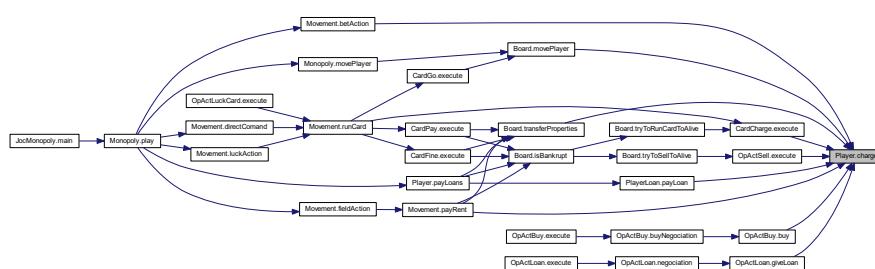
La quantitat entrada `amount` ha estat sumada de `money` del Jugador.

Paràmetres

<code>amount</code>	quantitat a cobrar
---------------------	--------------------

Definició a la línia 53 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.5 getBankruptcy()

```
Boolean Player.getBankruptcy ( )
```

Retorna si el Jugador es en fallida o no.

Precondició

true

Postcondició

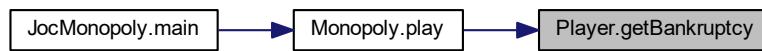
És retorna true si Jugador es en fallida, false altrament.

Retorna

true si bankruptcy es true, false altrament.

Definició a la línia 215 del fitxer Player.java.

Gràfic de crides a aquesta funció:

**5.26.3.6 getFields()**

```
ArrayList<BoxField> Player.getFields( )
```

Retorna la llista boxes_in_property de terrenys que te en propietat el Jugador.

Precondició

true

Postcondició

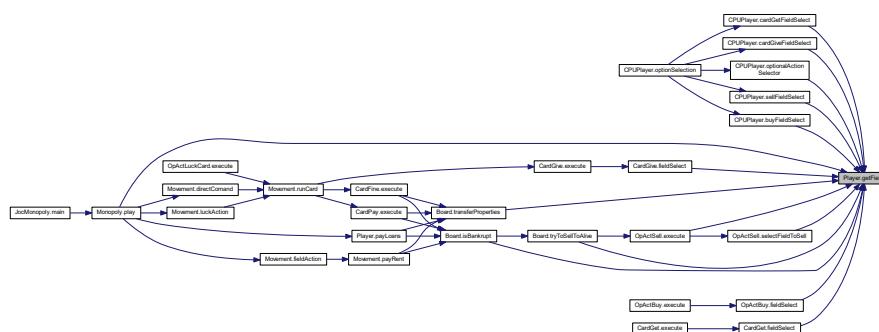
La llista boxes_in_property ha estat retornada.

Retorna

ArrayList que contindrà tots els terrenys en propeitat del Jugador.

Definició a la línia 181 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.7 getLoans()

```
ArrayList<PlayerLoan> Player.getLoans ( )
```

Definició a la línia 205 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.8 getLuckCards()

```
List<Card> Player.getLuckCards ( )
```

Retorna en una llista les targetes sort luckCards del Jugador.

Precondició

true

Postcondició

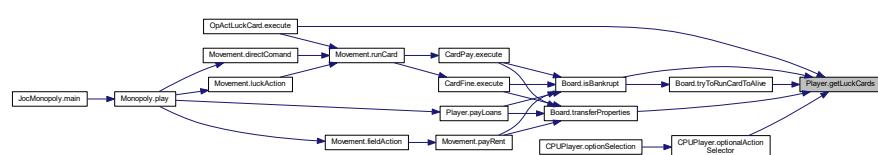
luckCards ha estat retornada.

Retorna

List de targetes sort del Jugador.

Definició a la línia 63 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.9 getMoney()

```
int Player.getMoney ( )
```

Retorna els diners que té el Jugador en possessió.

Precondició

true

Postcondició

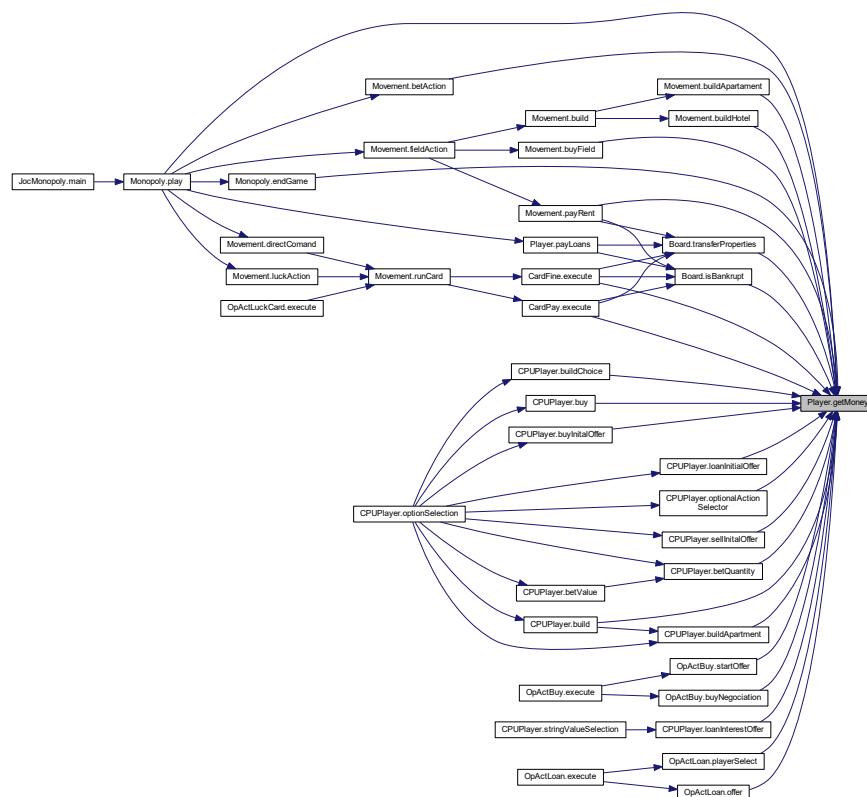
Els diners `money` del Jugador han estat retornats.

Retorna

int que contindrà els diners `money` de Jugador.

Definició a la línia 125 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.10 getName()

```
String Player.getName ( )
```

Retorna el nom del Jugador el qual la crida.

Precondició

true

Postcondició

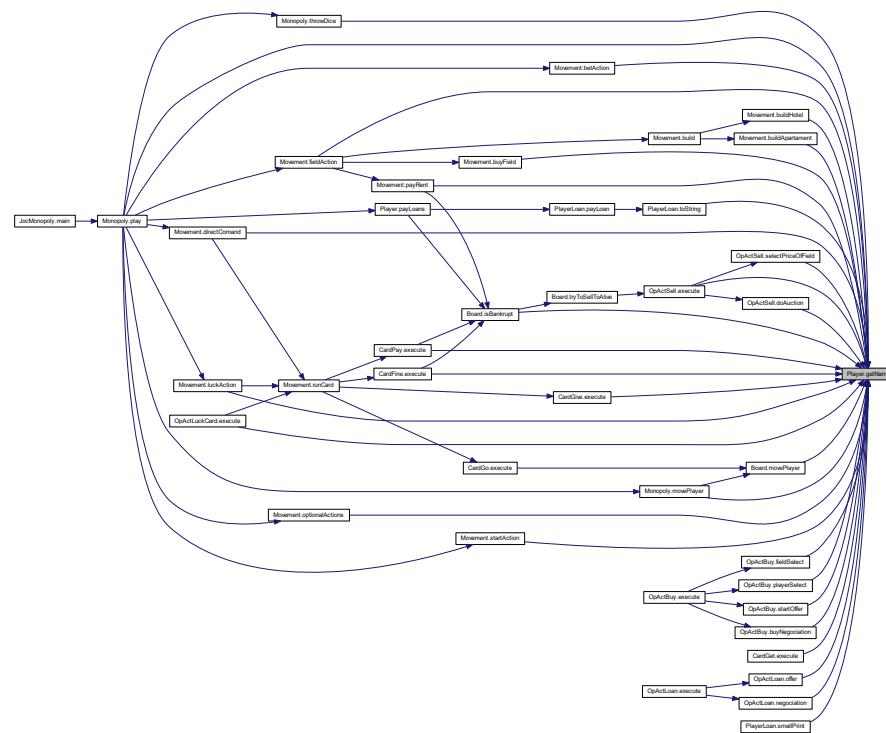
El nom del Jugador `name` ha estat tornat.

Retorna

String que contindrà el nom `name` del Jugador.

Definició a la línia 93 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.11 getPosition()

```
int Player.getPosition ( )
```

Retorna la posició del Jugador el qual la crida.

Precondició

true

Postcondició

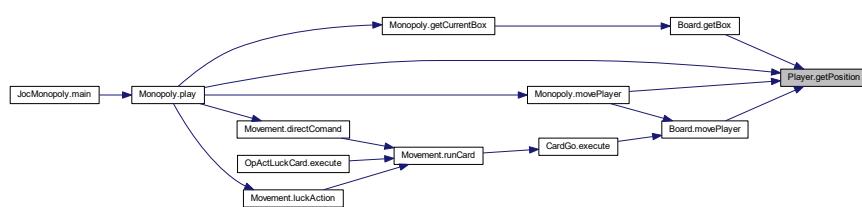
la posició del Jugador `position` ha estat tornada.

Retorna

int que contindrà la `position` del Jugador.

Definició a la línia 114 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.12 getType()

```
String Player.getType ( )
```

Retorna el tipus de Jugador el qual la crida.

Precondició

true

Postcondició

El tipus del Jugador `type` ha estat tornat.

Retorna

String que contindrà el tipus `type` del Jugador.

Definició a la línia 103 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.13 goToBankruptcy()

```
void Player.goToBankruptcy( )
```

Declara el Jugador en fallida.

Precondició

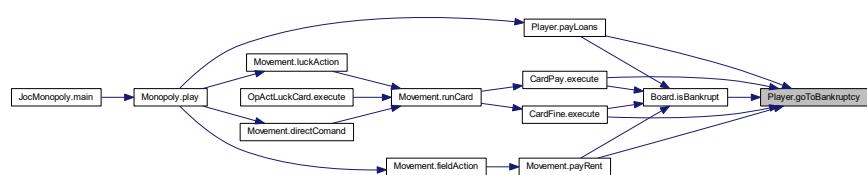
true

Postcondició

El jugador ha estat declarat en fallida.

Definició a la línia 191 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.14 haveFields()

```
boolean Player.haveFields( )
```

Retorna si el Jugador té terrenys en propietat o no.

Precondició

true

Postcondició

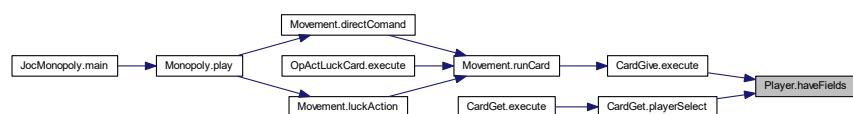
És retorna true si Jugador té terrenys, false altrament.

Retorna

true si `boxes_in_property` no es buida, false altrament.

Definició a la línia 201 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.15 movePlayer()

```
void Player.movePlayer (
    int position )
```

Mou el Jugador a una posició `position` final en el tauler.

Precondició

`position > 0`

Postcondició

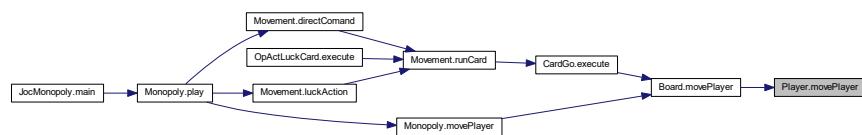
El Jugador ha sigut mogut a `position` del tauler.

Paràmetres

<code>position</code>	posició final a la qual el Jugador s'ha de moure.
-----------------------	---

Definició a la línia 147 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.16 numberOfAgrupationField()

```
int Player.numberOfAgrupationField (
    String group_name )
```

calcula el numero de terrenys de la agrupacio passada que te en propietat el jugador

Precondició

`true`

Postcondició

el numero de terrenys de la agrupacio passada s'ha retornat

Paràmetres

<i>group_name</i>	nom de la agrupació
-------------------	---------------------

Retorna

numero de terrenys de la agrupació passada

Definició a la línia 264 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.17 `numberOfBuildingsAffordable()`

```
int Player.numberOfBuildingsAffordable (
    int price )
```

Calcula el nombre de apartaments que es pot permetre costruir al preu entrat.

Precondició

`price > 0`

Postcondició

el nombre de apartaments construïbles ha estat retornada

Paràmetres

<i>price</i>	preu de un apartament
--------------	-----------------------

Retorna

int que contindrà el nombre de apartaments construïbles

Definició a la línia 136 del fitxer Player.java.

Gràfic de crides a aquesta funció:



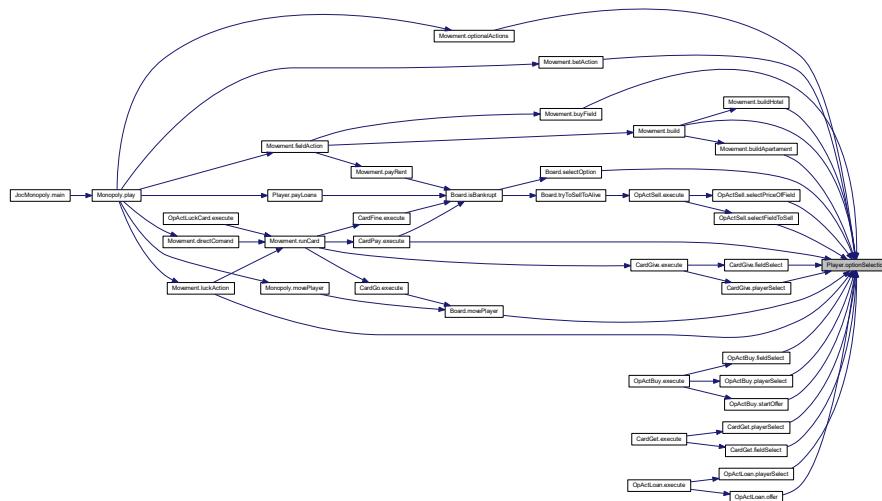
5.26.3.18 optionSelection()

```

abstract int Player.optionSelection (
    String type,
    Player player,
    BoxField field,
    ArrayList< Integer > options,
    ArrayList< Player > players,
    Card card,
    int value,
    ArrayList< optionalActions > optional_actions ) [abstract]
  
```

Reimplementat a [TerminalPlayer](#) i [CPUPlayer](#).

Gràfic de crides a aquesta funció:



5.26.3.19 pay()

```

void Player.pay (
    int amount )
  
```

El Jugador paga una quantitat de diners `amount` determinada.

Precondició

```
amount >= 0
```

Postcondició

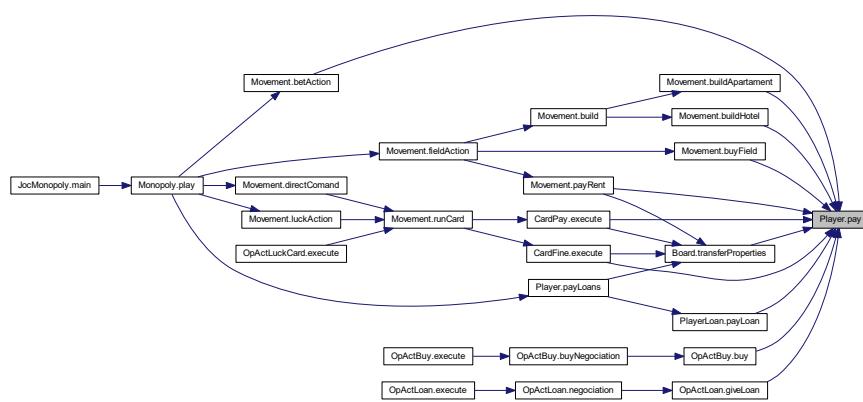
La quantitat entrada `amount` ha estat restada de `money` del Jugador.

Paràmetres

<code>amount</code>	quantitat a pagar
---------------------	-------------------

Definició a la línia 43 del fitxer Player.java.

Gràfic de crides a aquesta funció:

**5.26.3.20 payLoans()**

```
void Player.payLoans (
    Board board,
    Movement movement )
```

Paga els préstecs que te pends els Jugadors i gestiona els torns dels que encara no s'han de pagar.

Precondició

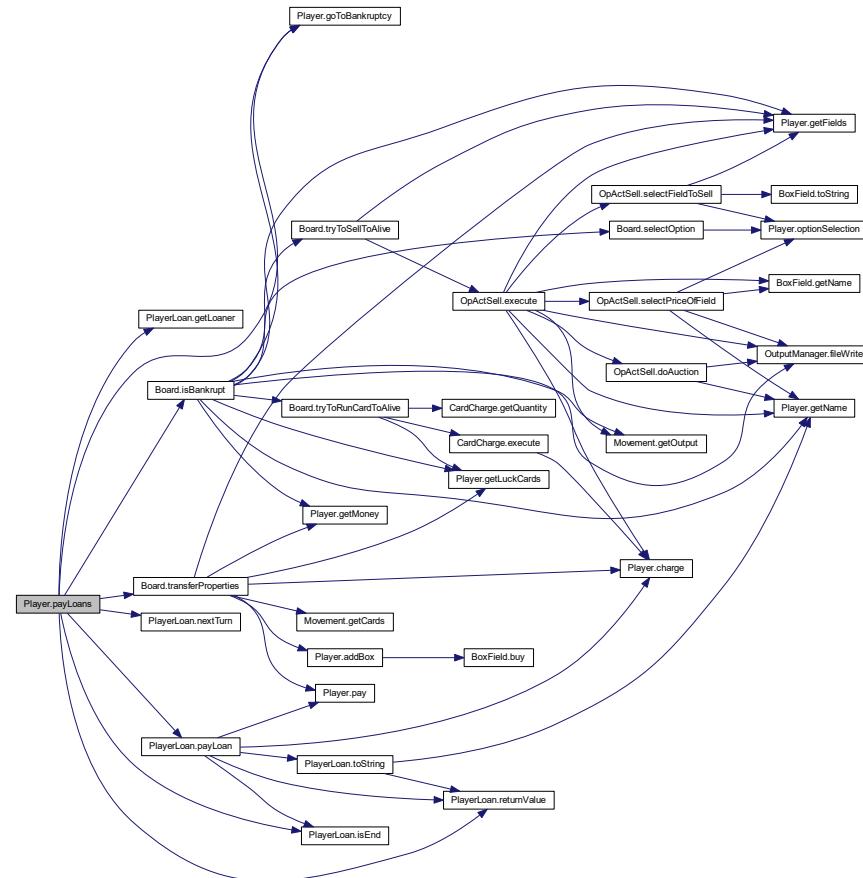
```
true
```

Postcondició

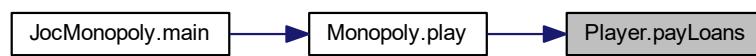
els préstecs que han finalitzat s'han retornat i gestionat els torns dels que no

Definició a la línia 238 del fitxer Player.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.26.3.21 removeBox()

```
void Player.removeBox (
    Box box )
```

Elimina un terreny a la propietat d'un Jugador.

Precondició

box != null

Postcondició

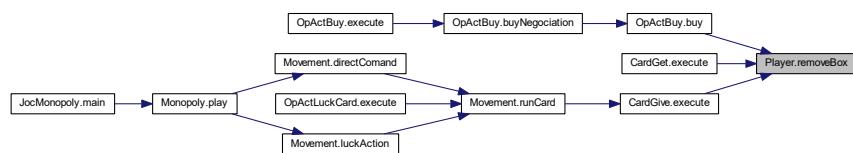
box ha estat eliminat de la llista de propietats boxes_in_property de Jugador.

Paràmetres

<i>box</i>	terreny que serà eliminat de la propietat de Jugador.
------------	---

Definició a la línia 171 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.3.22 removeLuckCard()

```
void Player.removeLuckCard (
    Card card )
```

Elimina una targeta sort card de la propietat del Jugador.

Precondició

card != null

Postcondició

La targeta card ha estat eliminada de la llista de targetes sort del Jugador.

Paràmetres

<code>card</code>	targeta que es vol eliminar de les targetes de sort <code>luckCards</code> del Jugador.
-------------------	---

Definició a la línia 83 del fitxer Player.java.

Gràfic de crides a aquesta funció:

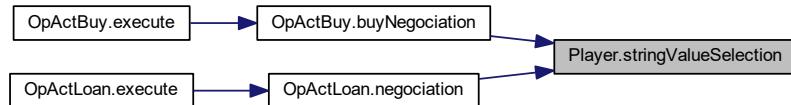
**5.26.3.23 stringValueSelection()**

```

abstract String Player.stringValueSelection (
    String type,
    Player player,
    BoxField field,
    int value,
    int second_value ) [abstract]
  
```

Reimplementat a [TerminalPlayer](#) i [CPUPlayer](#).

Gràfic de crides a aquesta funció:



5.26.3.24 `toString()`

```
String Player.toString ( )
```

toString per mostrar l'informació de [Player](#) per text.

Precondició

true

Postcondició

[Player](#) ha estat mostrat per pantalla amb tota la seva informació per poder seguir la partida adequadament.

Retorna

String buida.

Definició a la línia 279 del fitxer Player.java.

Gràfic de crides a aquesta funció:



5.26.4 Documentació de les Dades Membre

5.26.4.1 `bankruptcy`

```
boolean Player.bankruptcy = false [private]
```

Estat fallida jugador.

Reimplementat a [CPUPlayer](#).

Definició a la línia 17 del fitxer Player.java.

5.26.4.2 boxes_in_property

```
final ArrayList<BoxField> Player.boxes_in_property = new ArrayList<>() [private]
```

Llista terrenys en propietat del Jugador.

Definició a la línia 14 del fitxer Player.java.

5.26.4.3 loans

```
ArrayList<PlayerLoan> Player.loans = new ArrayList<>() [private]
```

Llista de préstecs del Jugador.

Definició a la línia 18 del fitxer Player.java.

5.26.4.4 luckCards

```
ArrayList<Card> Player.luckCards = new ArrayList<>() [private]
```

Llista targetes sort.

Definició a la línia 16 del fitxer Player.java.

5.26.4.5 money

```
int Player.money [private]
```

Diners del Jugador.

Definició a la línia 13 del fitxer Player.java.

5.26.4.6 name

```
final String Player.name [private]
```

Nom del Jugador.

Definició a la línia 12 del fitxer Player.java.

5.26.4.7 position

```
int Player.position [private]
```

Posició del Jugador.

Definició a la línia 15 del fitxer Player.java.

5.26.4.8 type

```
final String Player.type [private]
```

Tipus de jugador (CPU, USER)

Definició a la línia 19 del fitxer Player.java.

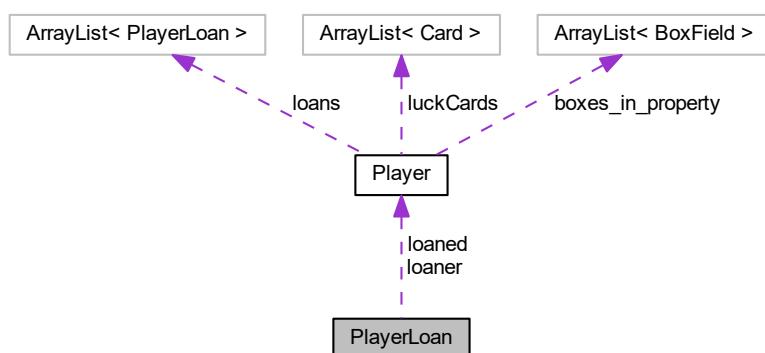
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [Player.java](#)

5.27 Referència de la Classe PlayerLoan

Classe que administra els préstecs entre prestatari i prestador.

Diagrama de col·laboració per a PlayerLoan:



Mètodes públics

- **PlayerLoan** (*Player* loaner, *Player* loaned, int value, int interest, int turns)

Constructor de PlayerLoan.
- void **nextTurn** ()

Resta un torn per haver de tornar el préstec.
- int **returnValue** ()

Calcula el valor que se l'hi haura de tornar al prestador.
- boolean **isEnd** ()

Consultor per saber si s'ha acabat els torns del préstec.
- boolean **payLoan** ()

Acció per pagar el préstec del loaned al loaner.
- String **smallPrint** ()

Sortida resumida dels préstecs actius de loaned.
- **Player getLoaner** ()

Sortida resumida dels préstecs actius de loaned.
- String **toString** ()

toString per mostrar la descripció dels préstecs PlayerLoan per text.

5.27.1 Descripció Detallada

Classe que administra els préstecs entre prestatari i prestador.

Definició a la línia 7 del fitxer PlayerLoan.java.

5.27.2 Documentació del Constructor i el Destructor

5.27.2.1 PlayerLoan()

```
PlayerLoan.PlayerLoan (
    Player loaner,
    Player loaned,
    int value,
    int interest,
    int turns )
```

Constructor de PlayerLoan.

Precondició

loaner != null i loaned != null

Postcondició

S'ha creat un préstec amb els atributs entrats.

Definició a la línia 20 del fitxer PlayerLoan.java.

5.27.3 Documentació de les Funcions Membre

5.27.3.1 getLoaner()

```
Player PlayerLoan.getLoaner ( )
```

Sortida resumida dels préstecs actius de `loaned`.

Precondició

`true`

Postcondició

S'ha mostrat els préstec que queden actius del jugador `loaned`.

Retorna

string amb la informació del préstec.

Definició a la línia 92 del fitxer `PlayerLoan.java`.

Gràfic de crides a aquesta funció:



5.27.3.2 isEnd()

```
boolean PlayerLoan.isEnd ( )
```

Consultor per saber si s'ha acabat els torns del préstec.

Precondició

`true`

Postcondició

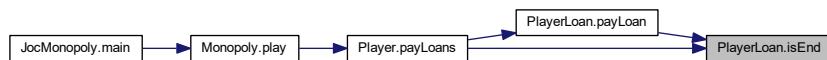
Retorna `true` si numero de torns finalitzat, `false` altrament

Retorna

`true` si `turns = 0`, `false` altrament.

Definició a la línia 54 del fitxer PlayerLoan.java.

Gràfic de crides a aquesta funció:



5.27.3.3 nextTurn()

```
void PlayerLoan.nextTurn ( )
```

Resta un torn per haver de tornar el préstec.

Precondició

`true`

Postcondició

Torn restat a `turns`.

Definició a la línia 33 del fitxer PlayerLoan.java.

Gràfic de crides a aquesta funció:



5.27.3.4 payLoan()

```
boolean PlayerLoan.payLoan ( )
```

Acció per pagar el préstec del loaned al loaner.

Precondició

true

Postcondició

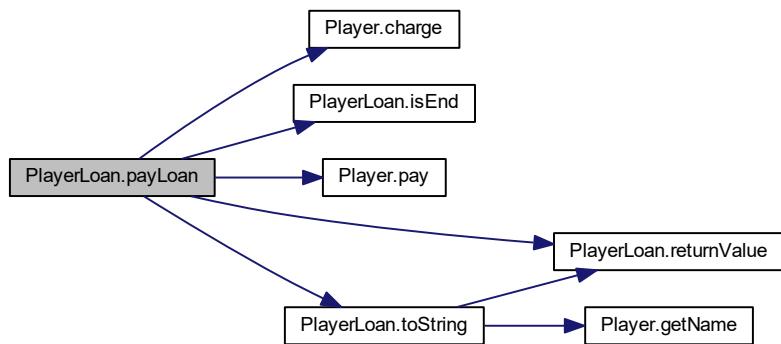
Retorna true si s'ha pagat el préstec, false altrament.

Retorna

true si s'ha pagat false altrament.

Definició a la línia 64 del fitxer PlayerLoan.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.27.3.5 returnValue()

```
int PlayerLoan.returnValue ( )
```

Calcula el valor que se l'hi haura de tornar al prestador.

Precondició

true

Postcondició

Retorne el valor a tornar al prestador.

Retorna

valor a retornar result.

Definició a la línia 43 del fitxer PlayerLoan.java.

Gràfic de crides a aquesta funció:



5.27.3.6 smallPrint()

```
String PlayerLoan.smallPrint ( )
```

Sortida resumida dels préstecs actius de loaned.

Precondició

true

Postcondició

S'ha mostrat els préstec que queden actius del jugador loaned.

Retorna

string amb la informació del préstec.

Definició a la línia 82 del fitxer PlayerLoan.java.

Gràfic de crides d'aquesta funció:



5.27.3.7 `toString()`

```
String PlayerLoan.toString ( )
```

`toString` per mostrar la descripció dels préstecs [PlayerLoan](#) per text.

Precondició

`true`

Postcondició

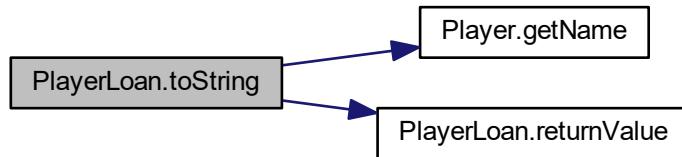
La descripció dels préstecs ha estat mostrada.

Retorna

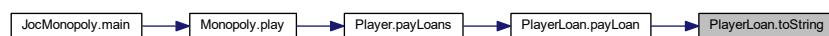
String buida.

Definició a la línia 103 del fitxer `PlayerLoan.java`.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [PlayerLoan.java](#)

5.28 Referència de la Classe TerminalPlayer

Jugador controlat per el usuari.

Diagrama d'Herència per a TerminalPlayer:

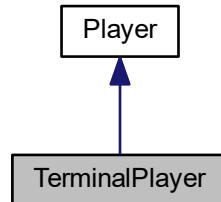
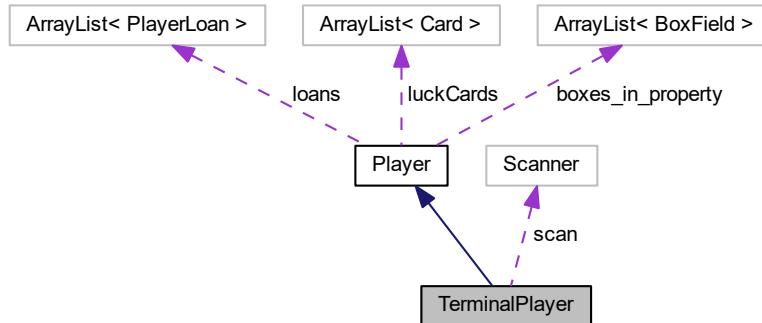


Diagrama de col·laboració per a TerminalPlayer:



Mètodes públics

- `TerminalPlayer (String name, int initial_money, int initial_position)`
Constructor de jugador amb terminal.
- `int optionSelection (String type, Player player, BoxField field, ArrayList< Integer > options, ArrayList< Player > players, Card card, int value, ArrayList< optionalActions > optional_actions)`
- `String stringValueSelection (String type, Player player, BoxField field, int value, int second_value)`

5.28.1 Descripció Detallada

Jugador controlat per el usuari.

Definició a la línia 11 del fitxer TerminalPlayer.java.

5.28.2 Documentació del Constructor i el Destructor

5.28.2.1 TerminalPlayer()

```
TerminalPlayer.TerminalPlayer (
    String name,
    int initial_money,
    int initial_position )
```

Constructor de jugador amb terminal.

Paràmetres

<i>name</i>	nom del Jugador.
<i>initial_money</i>	quantitat de diners inicials dels que disposa un Jugador.
<i>initial_position</i>	posició inicial del Jugador.

Precondició

true

Postcondició

Crea un jugador amb els atributs entrats.

Definició a la línia 22 del fitxer TerminalPlayer.java.

5.28.3 Documentació de les Funcions Membre

5.28.3.1 optionSelection()

```
int TerminalPlayer.optionSelection (
    String type,
    Player player,
    BoxField field,
    ArrayList< Integer > options,
    ArrayList< Player > players,
    Card card,
    int value,
    ArrayList< optionalActions > optional_actions )
```

Reimplementat de [Player](#).

Definició a la línia 26 del fitxer TerminalPlayer.java.

5.28.3.2 stringValueSelection()

```
String TerminalPlayer.stringValueSelection (
    String type,
    Player player,
    BoxField field,
    int value,
    int second_value )
```

Reimplementat de [Player](#).

Definició a la línia 41 del fitxer TerminalPlayer.java.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [TerminalPlayer.java](#)

5.29 Referència de la Classe UIFileSelector

Implementació d'un selector de fitxers gràfic per poder jugar al [Monopoly](#).

Diagrama d'Herència per a UIFileSelector:

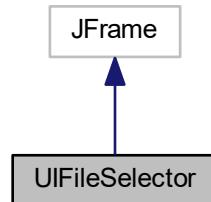
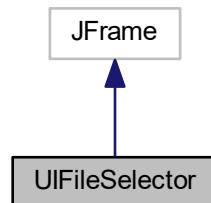


Diagrama de col·laboració per a UIFileSelector:



Mètodes públics

- `UIFileSelector ()`
Constructor de `UIFileSelector()`.
- `String getRulesFileName ()`
Getter de nom de fitxer regles `rules_name`.
- `String getBoardFileName ()`
Getter de nom de fitxer tauler `board_name`.
- `boolean getStatus ()`
Getter de estat de finestra `status`.

Mètodes Privats

- `void startComponents ()`
Inicialitza i crea tota la finestra.
- `void createLabels (JPanel panel)`
Crea etiquetes i les afegeix a la finestra.
- `void defineButtons (JButton browse_rules_button, JButton browse_board_button, JButton next_button, JPanel panel)`
Afegeix botons a la finestra.
- `void defineTextFields (JTextField rules_field, JTextField board_field, JPanel panel)`
Afegeix camps de text a la finestra.
- `void defineTextAreas (JTextArea rules_text_area, JTextArea board_text_area, JPanel panel)`
Afegeix areas de text a la finestra.
- `void createBrowseButtonAction (JButton browse_button, JTextField browse_field, JTextArea browse_textarea, String name_button, JPanel panel)`
Crea les accions pels botons browse.
- `void createNextButtonAction (JButton next_button)`
Crea les accions pel boto next.

Atributs Privats

- `String rules_name`
Ruta fitxer de regles.
- `String board_name`
Ruta fitxer de tauler.
- `boolean status = false`
Estat de selector de fitxers, true si s'ha seleccionat false altrament.

5.29.1 Descripció Detallada

Implementació d'un selector de fitxers gràfic per poder jugar al [Monopoly](#).

Definició a la línia 15 del fitxer `UIFileSelector.java`.

5.29.2 Documentació del Constructor i el Destructor

5.29.2.1 `UIFileSelector()`

```
UIFileSelector.UIFileSelector ( )
```

Constructor de [UIFileSelector\(\)](#).

Precondició

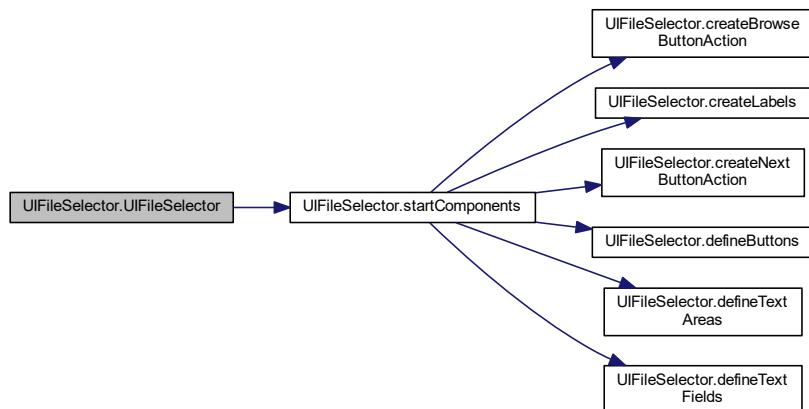
true

Postcondició

Crea un sel·lector de fitxers d'entrada.

Definició a la línia 26 del fitxer UIFileSelector.java.

Gràfic de crides d'aquesta funció:



5.29.3 Documentació de les Funcions Membre

5.29.3.1 `createBrowseButtonAction()`

```
void UIFileSelector.createBrowseButtonAction (
    JButton browse_button,
    JTextField browse_field,
    JTextArea browse_textarea,
    String name_button,
    JPanel panel ) [private]
```

Crea les accions pels botons browse.

Precondició

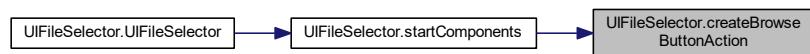
```
true
```

Postcondició

Accions creades per a cada boto de browse.

Definició a la línia 154 del fitxer UIFileSelector.java.

Gràfic de crides a aquesta funció:

**5.29.3.2 createLabels()**

```
void UIFileSelector.createLabels (
    JPanel panel ) [private]
```

Crea etiquetes i les afegeix a la finestra.

Precondició

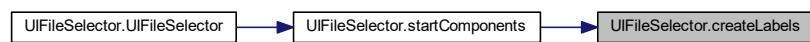
```
true
```

Postcondició

Etiquetes afegides a la finestra.

Definició a la línia 88 del fitxer UIFileSelector.java.

Gràfic de crides a aquesta funció:



5.29.3.3 createNextButtonAction()

```
void UIFileSelector.createNextButtonAction (
    JButton next_button ) [private]
```

Crea les accions pel boto next.

Precondició

true

Postcondició

Accions creades pel boto next.

Definició a la línia 191 del fitxer UIFileSelector.java.

Gràfic de crides a aquesta funció:



5.29.3.4 defineButtons()

```
void UIFileSelector.defineButtons (
    JButton browse_rules_button,
    JButton browse_board_button,
    JButton next_button,
    JPanel panel ) [private]
```

Afegeix botos a la finestra.

Precondició

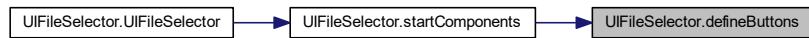
true

Postcondició

Botons definits a la finestra.

Definició a la línia 104 del fitxer UIFileSelector.java.

Gràfic de crides a aquesta funció:



5.29.3.5 defineTextAreas()

```
void UIFileSelector.defineTextAreas (
    JTextArea rules_text_area,
    JTextArea board_text_area,
    JPanel panel ) [private]
```

Afegeix areas de text a la finestra.

Precondició

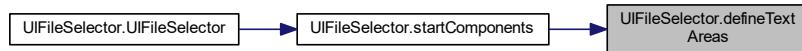
true

Postcondició

Les areas de text han estat definides.

Definició a la línia 135 del fitxer UIFileSelector.java.

Gràfic de crides a aquesta funció:



5.29.3.6 defineTextFields()

```
void UIFileSelector.defineTextFields (
    JTextField rules_field,
    JTextField board_field,
    JPanel panel ) [private]
```

Afegeix camps de text a la finestra.

Precondició

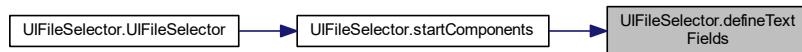
true

Postcondició

Els camps de text han estat definits.

Definició a la línia 123 del fitxer UIFileSelector.java.

Gràfic de crides a aquesta funció:



5.29.3.7 getBoardFileName()

```
String UIFileSelector.getBoardFileName ( )
```

Getter de nom de fitxer tauler `board_name`.

Precondició

`true`

Postcondició

el nom de fitxer `board_name` ha estat tornat.

Definició a la línia 72 del fitxer `UIFileSelector.java`.

Gràfic de crides a aquesta funció:



5.29.3.8 getRulesFileName()

```
String UIFileSelector.getRulesFileName ( )
```

Getter de nom de fitxer regles `rules_name`.

Precondició

`true`

Postcondició

el nom de fitxer `rules_name` ha estat tornat.

Definició a la línia 63 del fitxer `UIFileSelector.java`.

Gràfic de crides a aquesta funció:



5.29.3.9 getStatus()

```
boolean UIFileSelector.getStatus ( )
```

Getter de estat de finestra status.

Precondició

true

Postcondició

el estat de finestra status ha estat tornat.

Definició a la línia 81 del fitxer UIFileSelector.java.

Gràfic de crides a aquesta funció:



5.29.3.10 startComponents()

```
void UIFileSelector.startComponents ( ) [private]
```

Inicialitza i crea tota la finestra.

Precondició

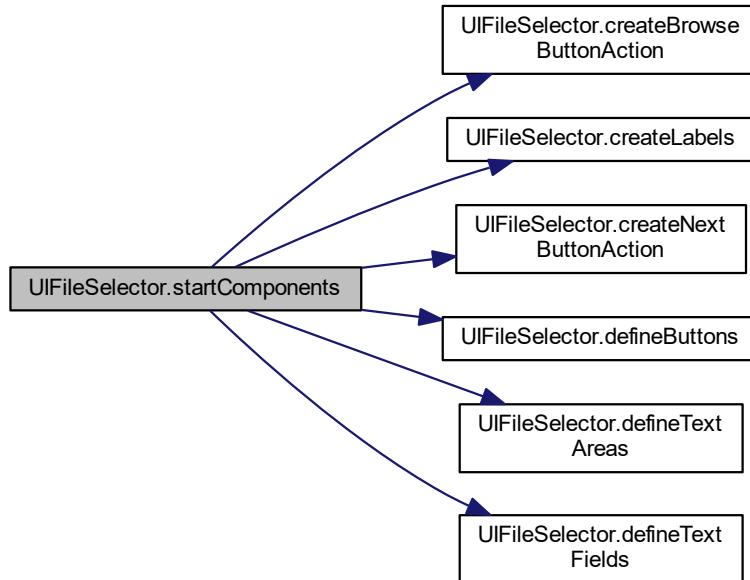
true

Postcondició

Finestra creada amb tota la seva intereficie gràfica.

Definició a la línia 38 del fitxer UIFileSelector.java.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



5.29.4 Documentació de les Dades Membre

5.29.4.1 board_name

```
String UIFileSelector.board_name [private]
```

Ruta fitxer de tauler.

Definició a la línia 18 del fitxer UIFileSelector.java.

5.29.4.2 **rules_name**

```
String UIFileSelector.rules_name [private]
```

Ruta fitxer de regles.

Definició a la línia 17 del fitxer `UIFileSelector.java`.

5.29.4.3 **status**

```
boolean UIFileSelector.status = false [private]
```

Estat de selector de fitxers, true si s'ha seleccionat false altrament.

Definició a la línia 19 del fitxer `UIFileSelector.java`.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- [UIFileSelector.java](#)

Capítol 6

Documentació dels Fitxers

6.1 Referència del Fitxer Board.java

Classes

- class [Board](#)

Classe del taulell. Gestiona la informació del taulell, les accions de moviment dels jugadors i altres com la gestió de la fallida.

6.1.1 Descripció Detallada

Autor

Gerard Rovellat

6.2 Referència del Fitxer Box.java

Classes

- class [Box](#)

Classe de Casella. Gestiona les dades i accions comunes entre tots els diferents tipus de casella.

6.2.1 Descripció Detallada

Autor

Marc Got

6.3 Referència del Fitxer BoxBet.java

Classes

- class [BoxBet](#)

6.4 Referència del Fitxer BoxDirectCommand.java

Classes

- class [BoxDirectCommand](#)

6.5 Referència del Fitxer BoxEmpty.java

Classes

- class [BoxEmpty](#)

6.6 Referència del Fitxer BoxField.java

Classes

- class [BoxField](#)

6.7 Referència del Fitxer BoxStart.java

Classes

- class [BoxStart](#)

6.8 Referència del Fitxer Card.java

Classes

- class [Card](#)

Implementa la funcions base i comunes totes les cartes.

6.8.1 Descripció Detallada

Autor

Gerard Rovellat

6.9 Referència del Fitxer CardCharge.java

Classes

- class [CardCharge](#)

Implementa la funcions de la carta de tipus cobrar.

6.9.1 Descripció Detallada

Autor

Marc Got

6.10 Referència del Fitxer CardFine.java

Classes

- class [CardFine](#)

Implementa la funcions de la carta de tipus multa.

6.10.1 Descripció Detallada

Autor

Marc Got

6.11 Referència del Fitxer CardGet.java

Classes

- class [CardGet](#)

Implementa la funcions de la carta de tipus rebre.

6.11.1 Descripció Detallada

Autor

Gerard Rovellat

6.12 Referència del Fitxer CardGive.java

Classes

- class [CardGive](#)

Implementa la funcions de la carta de tipus donar.

6.12.1 Descripció Detallada

Autor

Marc Got

6.13 Referència del Fitxer CardGo.java

Classes

- class [CardGo](#)

Targeta que mou al jugador a una casella determinada en el tauler de [Monopoly](#).

6.13.1 Descripció Detallada

Autor

Gerard Rovellat

6.14 Referència del Fitxer CardPay.java

Classes

- class [CardPay](#)

Targeta que fa pagar la quantitat assignada a un jugador.

6.14.1 Descripció Detallada

Autor

Marc Got

6.15 Referència del Fitxer CPUPlayer.java

Classes

- class [CPUPlayer](#)

Jugador controlat per el ordinador.

6.15.1 Descripció Detallada

Autor

Marc Got

6.16 Referència del Fitxer JocMonopoly.java

Classes

- class [JocMonopoly](#)

Main principal del joc del [Monopoly](#).

6.16.1 Descripció Detallada

Autor

Gerard Rovellat

6.17 Referència del Fitxer JSONManager.java

Classes

- class [JSONManager](#)

JSONManager administra totes les funcions relacionades amb llegir fitxers JSON i també la creació del fitxer de desenvolupament de la partida.

6.18 Referència del Fitxer Monopoly.java

Classes

- class [Monopoly](#)

Classe que s'encarrega d'administrar l'inicialització, els torns i funcionament general i la finalització del joc [Monopoly](#).

6.18.1 Descripció Detallada

Autor

Marc Got

6.19 Referència del Fitxer Movement.java

Classes

- class [Movement](#)

Classe que s'encarrega de gestionar les accions que es poden fer en un moviment. Aquest moviment sera fet sobre un Jugador.

6.19.1 Descripció Detallada

Autor

Marc Got

6.20 Referència del Fitxer OpActBuy.java

Classes

- class [OpActBuy](#)

6.21 Referència del Fitxer OpActLoan.java

Classes

- class [OpActLoan](#)

6.22 Referència del Fitxer OpActLuckCard.java

Classes

- class [OpActLuckCard](#)

6.23 Referència del Fitxer OpActSell.java

Classes

- class [OpActSell](#)

6.24 Referència del Fitxer optionalActions.java

Classes

- class [optionalActions](#)
Interfície general d'accions opcionals del joc de [Monopoly](#).

6.24.1 Descripció Detallada

Autor

Gerard Rovellat

6.25 Referència del Fitxer OutputManager.java

Classes

- class [OutputManager](#)
Classe que s'encarrega de gestionar les sortides en el nostre cas al fitxer de desenvolupament de partida.

6.25.1 Descripció Detallada

Autor

Gerard Rovellat

6.26 Referència del Fitxer Player.java

Classes

- class [Player](#)

Classe de Jugador del Monopoly. En aquesta classe hi haurà tota la informació del jugador i algunes accions que pot fer el jugador en si.

6.26.1 Descripció Detallada

Autor

Marc Got

6.27 Referència del Fitxer PlayerLoan.java

Classes

- class [PlayerLoan](#)

Classe que administra els préstecs entre prestatari i prestador.

6.27.1 Descripció Detallada

Autor

Marc Got

6.28 Referència del Fitxer README.md

6.29 Referència del Fitxer TerminalPlayer.java

Classes

- class [TerminalPlayer](#)

Jugador controlat per el usuari.

6.29.1 Descripció Detallada

Autor

Marc Got

6.30 Referència del Fitxer UIFileSelector.java

Classes

- class [UIFileSelector](#)
Implementació d'un selector de fitxers gràfic per poder jugar al Monopoly.

6.30.1 Descripció Detallada

Autor

Gerard Rovellat

Índex alfabètic

active_player
 Movement, 161
addBox
 Board, 11
 Player, 196
addLoan
 Player, 197
addLuckCard
 Player, 198
addPlayer
 Board, 12
askInfoOfPlayers
 Monopoly, 130
askNrOfPlayers
 Monopoly, 131

bankruptcy
 CPUPlayer, 92
 Player, 213
basic_rent
 BoxField, 53
betAction
 Movement, 146
betQuantity
 CPUPlayer, 93
betResult
 BoxBet, 34
betValue
 CPUPlayer, 94
Board, 9
 addBox, 11
 addPlayer, 12
 Board, 10
 board, 26
 boxes_nr, 26
 getBox, 13
 getSize, 13
 haveAvailableFields, 14
 haveOwner, 15
 isBankrupt, 16
 movePlayer, 17
 numberOfAgrupationField, 19
 players, 27
 randomField, 19
 selectOption, 20
 setBoxesNr, 21
 toString, 22
 transferProperties, 22
 tryToRunCardToAlive, 24
 tryToSellToAlive, 25
board
 Board, 26
 Monopoly, 141
Board.java, 235
board_file
 JSONManager, 126
board_name
 UIFileSelector, 232
bought
 BoxField, 53
Box, 27
 Box, 28
 getPosition, 29
 getType, 29
 name, 32
 playerInPosition, 30
 position, 32
 print, 31
 type, 32
Box.java, 235
BoxBet, 33
 betResult, 34
 BoxBet, 34
 combinations, 35
BoxBet.java, 235
BoxDirectCommand, 35
 BoxDirectCommand, 36
 function, 38
 getCard, 37
BoxDirectCommand.java, 236
BoxEmpty, 38
 BoxEmpty, 39
BoxEmpty.java, 236
boxes_in_property
 Player, 213
boxes_nr
 Board, 26
BoxField, 40
 basic_rent, 53
 bought, 53
 BoxField, 42
 build, 43
 buildable, 53
 builded, 54
 building_price, 54
 buildings_rent, 54
 buy, 43
 getGroup, 44

getName, 44
 getNumberOfApartaments, 45
 getNumberOfHotels, 46
 getOwner, 46
 getPrice, 47
 getRent, 48
 group, 54
 group_rent, 54
 hotel, 55
 hotel_price, 55
 hotel_rent, 55
 hotelBuildable, 48
 houseBuildable, 49
 houseBuildableType, 49
 isBought, 50
 max_buildings, 55
 name, 55
 number_ofHouseBuildable, 50
 owner, 56
 price, 56
 priceToBuild, 51
 sell, 52
 toString, 52
BoxField.java, 236
BoxStart, 56

- BoxStart**, 57
- getType, 58

BoxStart.java, 236
build

- BoxField**, 43
- CPUPlayer**, 95
- Movement**, 147

buildable

- BoxField**, 53

buildApartment

- Movement**, 148

buildApartment

- CPUPlayer**, 96

buildChoice

- CPUPlayer**, 97

builded

- BoxField**, 54

buildHotel

- Movement**, 149

building_price

- BoxField**, 54

buildings_rent

- BoxField**, 54

buy

- BoxField**, 43
- CPUPlayer**, 98
- OpActBuy**, 163

buyBuyerOffer

- CPUPlayer**, 99

buyField

- Movement**, 150

buyFieldSelect

- CPUPlayer**, 100

buyInitialOffer

- CPUPlayer**, 100

buyNegociation

- OpActBuy**, 164

buyPlayerSelect

- CPUPlayer**, 101

buySellerOffer

- CPUPlayer**, 102

Card, 58

- Card**, 59
- getType, 60
- isPostposable, 60
- postposable, 61
- type, 61

Card.java, 236
CardCharge, 62

- CardCharge**, 63
- execute, 63
- getQuantity, 64
- message, 65
- messages, 66
- quantity, 66
- toString, 65

CardCharge.java, 236
CardFine, 66

- CardFine**, 67
- execute, 68
- message, 70
- messages, 70
- quantity, 70
- toString, 69

CardFine.java, 237
CardGet, 71

- CardGet**, 72
- execute, 72
- fieldSelect, 73
- playerSelect, 74
- toString, 75

CardGet.java, 237
cardGetFieldSelect

- CPUPlayer**, 103

cardGetPlayerSelect

- CPUPlayer**, 104

CardGive, 76

- CardGive**, 77
- execute, 78
- fieldSelect, 79
- playerSelect, 80
- toString, 81

CardGive.java, 237
cardGiveFieldSelect

- CPUPlayer**, 105

cardGivePlayerSelect

- CPUPlayer**, 106

CardGo, 81

- CardGo**, 82
- execute, 83
- position, 85

toString, 84
CardGo.java, 238
CardPay, 85
 CardPay, 86
 execute, 87
 quantity, 89
 toString, 88
CardPay.java, 238
cardPayPlayerSelect
 CPUPlayer, 106
cards
 JSONManager, 126
 Monopoly, 141
 Movement, 161
charge
 Player, 199
checkEndGame
 Monopoly, 131
combinations
 BoxBet, 35
CPUPlayer, 89
 bankruptcy, 92
 betQuantity, 93
 betValue, 94
 build, 95
 buildApartment, 96
 buildChoice, 97
 buy, 98
 buyBuyerOffer, 99
 buyFieldSelect, 100
 buyInitialOffer, 100
 buyPlayerSelect, 101
 buySellerOffer, 102
 cardGetFieldSelect, 103
 cardGetPlayerSelect, 104
 cardGiveFieldSelect, 105
 cardGivePlayerSelect, 106
 cardPayPlayerSelect, 106
 CPUPlayer, 91
 loanInitialOffer, 107
 loanInterestOffer, 108
 loanPlayerSelect, 109
 loanTurnsOffer, 110
 optionalActionSelector, 110
 optionSelection, 111
 postposableLuckCardChoice, 113
 rand, 119
 sellBuyerOffer, 114
 sellFieldSelect, 115
 sellInitialOffer, 116
 start, 117
 stringValueSelection, 118
CPUPlayer.java, 238
createBrowseButtonAction
 UIFileSelector, 226
createLabels
 UIFileSelector, 227
createNextButtonAction

 UIFileSelector, 227
current_box
 Movement, 161
current_player
 Monopoly, 142
current_player_iterator
 Monopoly, 142
defineButtons
 UIFileSelector, 228
defineTextAreas
 UIFileSelector, 228
defineTextFields
 UIFileSelector, 229
dev_file
 OutputManager, 193
dice_result
 Monopoly, 142
directComand
 Movement, 151
doAuction
 OpActSell, 183
endGame
 Monopoly, 132
execute
 CardCharge, 63
 CardFine, 68
 CardGet, 72
 CardGive, 78
 CardGo, 83
 CardPay, 87
 OpActBuy, 166
 OpActLoan, 172
 OpActLuckCard, 180
 OpActSell, 185
 optionalActions, 190
fieldAction
 Movement, 152
fieldSelect
 CardGet, 73
 CardGive, 79
 OpActBuy, 167
fileWrite
 OutputManager, 192
function
 BoxDirectCommand, 38
getBankruptcy
 Player, 199
getBoardFileName
 UIFileSelector, 229
getBox
 Board, 13
getCard
 BoxDirectCommand, 37
getCards
 Movement, 154

getCurrentBox
 Monopoly, 133
 getFields
 Player, 200
 getGroup
 BoxField, 44
 getLoaner
 PlayerLoan, 217
 getLoans
 Player, 200
 getLuckCards
 Player, 201
 getMode
 Monopoly, 134
 getMoney
 Player, 201
 getName
 BoxField, 44
 Player, 202
 getNumberOfApartaments
 BoxField, 45
 getNumberOfHotels
 BoxField, 46
 getOutput
 Movement, 154
 getOwner
 BoxField, 46
 getPosition
 Box, 29
 Player, 203
 getPrice
 BoxField, 47
 getQuantity
 CardCharge, 64
 getRent
 BoxField, 48
 getRulesFileName
 UIFileSelector, 230
 getSize
 Board, 13
 getStatus
 UIFileSelector, 230
 getType
 Box, 29
 BoxStart, 58
 Card, 60
 Player, 204
 giveLoan
 OpActLoan, 173
 goToBankruptcy
 Player, 204
 group
 BoxField, 54
 group_rent
 BoxField, 54
 haveAvailableFields
 Board, 14
 haveFields

 Player, 205
 haveOwner
 Board, 15
 hotel
 BoxField, 55
 hotel_price
 BoxField, 55
 hotel_rent
 BoxField, 55
 hotelBuildable
 BoxField, 48
 houseBuildable
 BoxField, 49
 houseBuildableType
 BoxField, 49
 initial_money
 JSONManager, 126
 Monopoly, 142
 isBankrupt
 Board, 16
 isBought
 BoxField, 50
 isEnd
 PlayerLoan, 217
 isPostposable
 Card, 60
 JocMonopoly, 119
 main, 120
 printHelp, 121
 JocMonopoly.java, 238
 JSONManager, 122
 board_file, 126
 cards, 126
 initial_money, 126
 JSONManager, 123
 mode, 127
 readBoard, 124
 readFile, 124
 readRules, 125
 rules_file, 127
 star_rewards, 127
 JSONManager.java, 239
 loanInitialOffer
 CPUPlayer, 107
 loanInterestOffer
 CPUPlayer, 108
 loanPlayerSelect
 CPUPlayer, 109
 loans
 Player, 214
 loanTurnsOffer
 CPUPlayer, 110
 luckAction
 Movement, 155
 luckCards
 Player, 214

main
 JocMonopoly, 120
max_buildings
 BoxField, 55
message
 CardCharge, 65
 CardFine, 70
messages
 CardCharge, 66
 CardFine, 70
mode
 JSONManager, 127
 Monopoly, 142
money
 Player, 214
Monopoly, 128
 askInfoOfPlayers, 130
 askNrOfPlayers, 131
 board, 141
 cards, 141
 checkEndGame, 131
 current_player, 142
 current_player_iterator, 142
 dice_result, 142
 endGame, 132
 getCurrentBox, 133
 getMode, 134
 initial_money, 142
 mode, 142
 Monopoly, 129
 movePlayer, 135
 nextPlayer, 136
 optional_actions, 143
 play, 137
 players, 143
 printPlayers, 139
 start_rewards, 143
 startGame, 139
 throwDice, 140
Monopoly.java, 239
Movement, 144
 active_player, 161
 betAction, 146
 build, 147
 buildApartment, 148
 buildHotel, 149
 buyField, 150
 cards, 161
 current_box, 161
 directComand, 151
 fieldAction, 152
 getCards, 154
 getOutput, 154
 luckAction, 155
 Movement, 145
 optionalActions, 156
 output, 161
 payRent, 157
 runCard, 158
 start_rewards, 161
 startAction, 160
Movement.java, 239
movePlayer
 Board, 17
 Monopoly, 135
 Player, 205
name
 Box, 32
 BoxField, 55
 Player, 214
negociation
 OpActLoan, 174
nextPlayer
 Monopoly, 136
nextTurn
 PlayerLoan, 218
numberOfAgrupationField
 Board, 19
 Player, 206
numberOfBuildingsAffordable
 Player, 207
numberOfHouseBuildable
 BoxField, 50
offer
 OpActLoan, 176
OpActBuy, 162
 buy, 163
 buyNegociation, 164
 execute, 166
 fieldSelect, 167
 OpActBuy, 163
 playerSelect, 168
 startOffer, 169
 toString, 170
OpActBuy.java, 239
OpActLoan, 171
 execute, 172
 giveLoan, 173
 negociation, 174
 offer, 176
 OpActLoan, 172
 playerSelect, 177
 toString, 178
OpActLoan.java, 240
OpActLuckCard, 179
 execute, 180
 OpActLuckCard, 180
 toString, 181
OpActLuckCard.java, 240
OpActSell, 182
 doAuction, 183
 execute, 185
 OpActSell, 183
 selectFieldToSell, 186
 selectPriceOfField, 187

toString, 188
 OpActSell.java, 240
 optional_actions
 Monopoly, 143
 optionalActions, 189
 execute, 190
 Movement, 156
 toString, 190
 optionalActions.java, 240
 optionalActionSelector
 CPUPlayer, 110
 optionSelection
 CPUPlayer, 111
 Player, 208
 TerminalPlayer, 223
 output
 Movement, 161
 OutputManager, 191
 dev_file, 193
 fileWrite, 192
 OutputManager, 191
 OutputManager.java, 240
 owner
 BoxField, 56

 pay
 Player, 208
 payLoan
 PlayerLoan, 218
 payLoans
 Player, 209
 payRent
 Movement, 157
 play
 Monopoly, 137
 Player, 193
 addBox, 196
 addLoan, 197
 addLuckCard, 198
 bankruptcy, 213
 boxes_in_property, 213
 charge, 199
 getBankruptcy, 199
 getFields, 200
 getLoans, 200
 getLuckCards, 201
 getMoney, 201
 getName, 202
 getPosition, 203
 getType, 204
 goToBankruptcy, 204
 haveFields, 205
 loans, 214
 luckCards, 214
 money, 214
 movePlayer, 205
 name, 214
 numberOfAgrupationField, 206
 numberOfBuildingsAffordable, 207

 optionSelection, 208
 pay, 208
 payLoans, 209
 Player, 196
 position, 214
 removeBox, 210
 removeLuckCard, 211
 stringValueSelection, 212
 toString, 212
 type, 215
 Player.java, 241
 playerInPosition
 Box, 30
 PlayerLoan, 215
 getLoaner, 217
 isEnd, 217
 nextTurn, 218
 payLoan, 218
 PlayerLoan, 216
 returnValue, 219
 smallPrint, 220
 toString, 220
 PlayerLoan.java, 241
 players
 Board, 27
 Monopoly, 143
 playerSelect
 CardGet, 74
 CardGive, 80
 OpActBuy, 168
 OpActLoan, 177
 position
 Box, 32
 CardGo, 85
 Player, 214
 postposable
 Card, 61
 postposableLuckCardChoice
 CPUPlayer, 113
 price
 BoxField, 56
 priceToBuild
 BoxField, 51
 print
 Box, 31
 printHelp
 JocMonopoly, 121
 printPlayers
 Monopoly, 139

 quantity
 CardCharge, 66
 CardFine, 70
 CardPay, 89

 rand
 CPUPlayer, 119
 randomField
 Board, 19

readBoard
 JSONManager, 124

readFile
 JSONManager, 124

README.md, 241

readRules
 JSONManager, 125

removeBox
 Player, 210

removeLuckCard
 Player, 211

returnValue
 PlayerLoan, 219

rules_file
 JSONManager, 127

rules_name
 UIFileSelector, 232

runCard
 Movement, 158

selectFieldToSell
 OpActSell, 186

selectOption
 Board, 20

selectPriceOfField
 OpActSell, 187

sell
 BoxField, 52

sellBuyerOffer
 CPUPlayer, 114

sellFieldSelect
 CPUPlayer, 115

sellInitialOffer
 CPUPlayer, 116

setBoxesNr
 Board, 21

smallPrint
 PlayerLoan, 220

star_rewards
 JSONManager, 127

start
 CPUPlayer, 117

start_rewards
 Monopoly, 143
 Movement, 161

startAction
 Movement, 160

startComponents
 UIFileSelector, 231

startGame
 Monopoly, 139

startOffer
 OpActBuy, 169

status
 UIFileSelector, 233

stringValueSelection
 CPUPlayer, 118
 Player, 212
 TerminalPlayer, 223

TerminalPlayer, 222
 optionSelection, 223
 stringValueSelection, 223
 TerminalPlayer, 223

TerminalPlayer.java, 241

throwDice
 Monopoly, 140

toString
 Board, 22
 BoxField, 52
 CardCharge, 65
 CardFine, 69
 CardGet, 75
 CardGive, 81
 CardGo, 84
 CardPay, 88
 OpActBuy, 170
 OpActLoan, 178
 OpActLuckCard, 181
 OpActSell, 188
 optionalActions, 190
 Player, 212
 PlayerLoan, 220

transferProperties
 Board, 22

tryToRunCardToAlive
 Board, 24

tryToSellToAlive
 Board, 25

type
 Box, 32
 Card, 61
 Player, 215

UIFileSelector, 224
 board_name, 232
 createBrowseButtonAction, 226
 createLabels, 227
 createNextButtonAction, 227
 defineButtons, 228
 defineTextAreas, 228
 defineTextFields, 229
 getBoardFileName, 229
 getRulesFileName, 230
 getStatus, 230
 rules_name, 232
 startComponents, 231
 status, 233
 UIFileSelector, 225

UIFileSelector.java, 242