ARCHITEKTUR

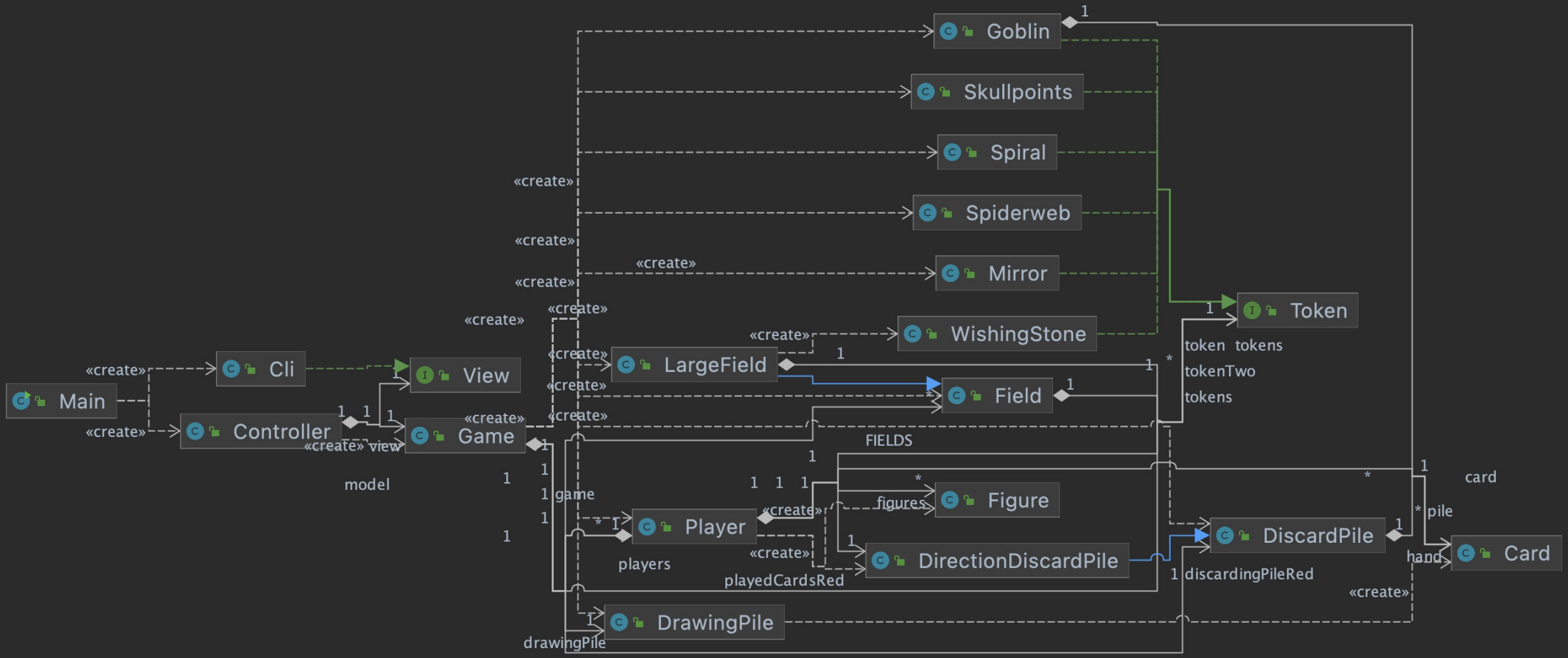
Marten Buchmann, Katinka Feltes, Carl Gathmann, Helen Kuswik

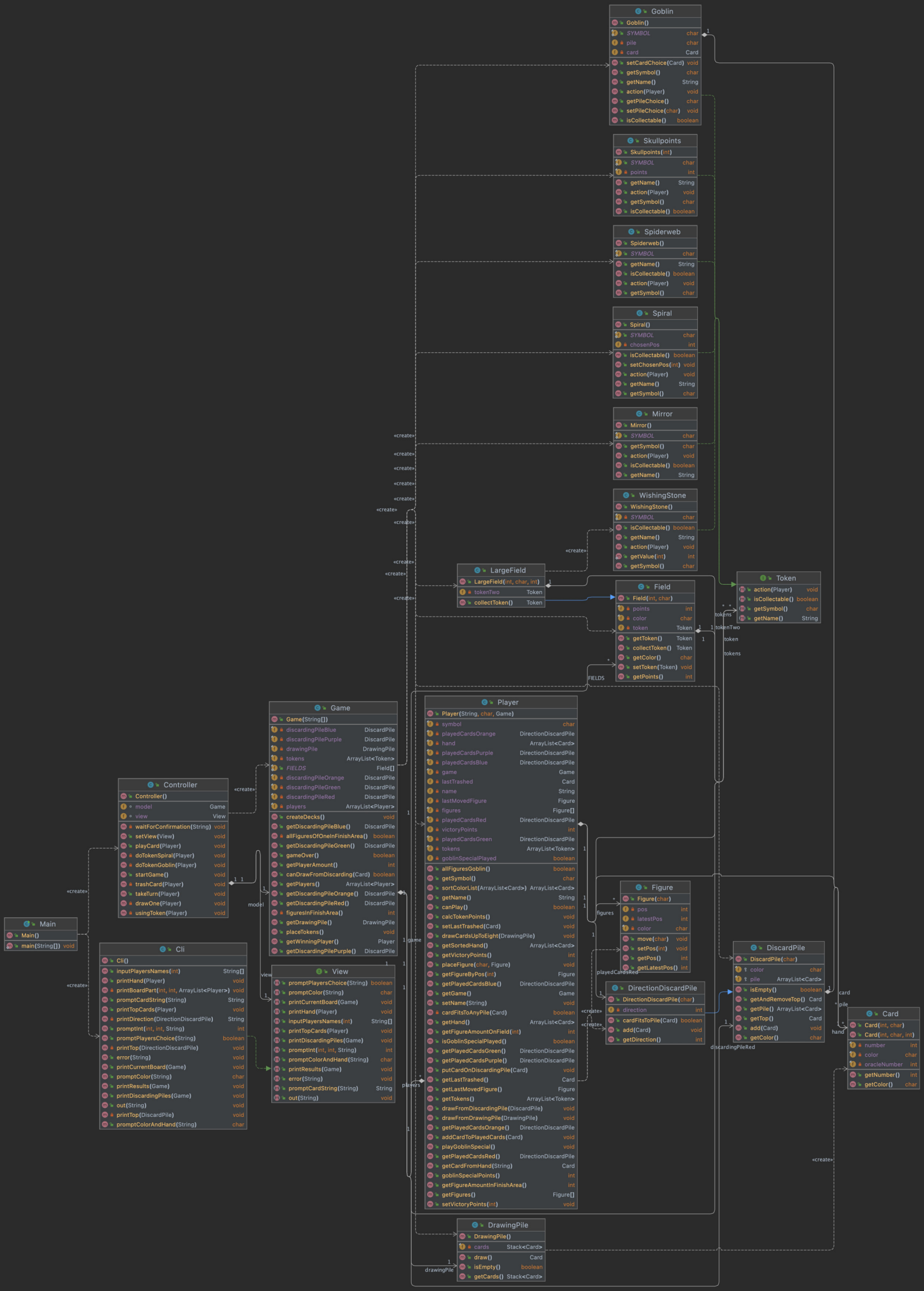
Erläuterungen zur Architektur

- Model-View-Controller gewählt
  → großer Vorteil: View ist nun komplett unabhängig von der Anzeige im CLI/GUI → vereinfacht es, später die GUI zu implementieren
  → Controller regelt den Spielablauf
  → Model entspricht der Klasse Game (und weiteren Klassen, auf die Game zugreift), in der die Funktionalität des Spiels implementiert ist
- Exceptions werden bei uns spezifischer ausgegeben:
  → „!Invalid move“:
    - falsche Zahl: „Please enter a number between x and y“
    - keine Zahl: „Input must be an Integer“
    - falsche Karte: „There is no card greater than 10“ / „Please enter one of the options above“
    - Zustimmung: „Please enter either y or n“
    - Eingabe von Stapel/Hand: „Please enter either r,g,b,o or p“ / „Please enter either r,g,b,o,p or h (hand)“
  → „!Move not allowed“:
    - Karte ziehen: „You fool: this pile is empty“ / „You can‘t draw a card you just trashed“
    - Karte spielen: „Card is not in hand“ / „Card does not fit into pile“ / „This figure can‘t move this far“
    - Karte abwerfen: „Card is not in hand“ / „Card is null or different color than the pile“
    - Spiral-Token: „You can not go to the field you came from“
    - Goblin-Token: „Card is not in hand“
- Tokens als Interface implementiert, damit dem Spielfeld generell Tokens zugeordnet werden können und weil sie alle die gleichen Grundfunktionen (Action, Name, collectable) haben

Klassendiagramm

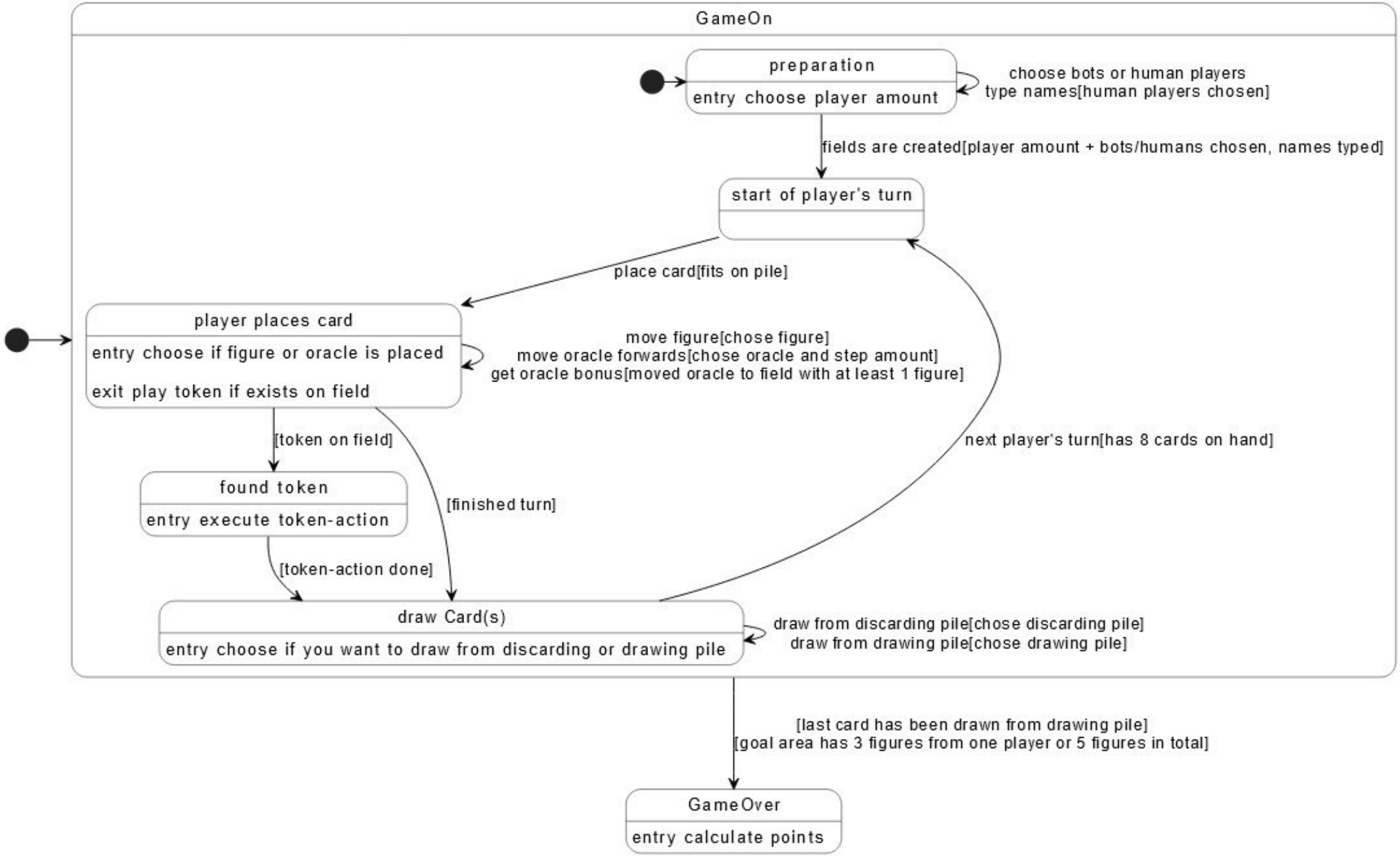
Gruppe 5


Goblin
Skullpoints
Spiral
Spiderweb
Mirror
WishingStone
Token
Cli
View
Main
Controller
Game
LargeField
Field
Player
Figure
DirectionDiscardPile
DiscardPile
Card
DrawingPile
«create»
model
game
players
figures
FIELDS
playedCardsRed
discardingPileRed
drawingPile
token
tokens
tokenTwo
pile
hand
card


Goblin
Goblin()
SYMBOL char
pile char
card Card
setCardChoice(Card) void
getSymbol() char
getName() String
action(Player) void
getPileChoice() char
setPileChoice(char) void
isCollectable() boolean
Skullpoints
Skullpoints(int)
SYMBOL char
points int
getName() String
action(Player) void
getSymbol() char
isCollectable() boolean
Spiderweb
Spiderweb()
SYMBOL char
getName() String
isCollectable() boolean
action(Player) void
getSymbol() char
Spiral
Spiral()
SYMBOL char
chosenPos int
isCollectable() boolean
setChosenPos(int) void
action(Player) void
getName() String
getSymbol() char
Mirror
Mirror()
SYMBOL char
getSymbol() char
action(Player) void
isCollectable() boolean
getName() String
WishingStone
WishingStone()
SYMBOL char
isCollectable() boolean
getName() String
action(Player) void
getValue(int) int
getSymbol() char
LargeField
LargeField(int, char, int)
tokenTwo Token
collectToken() Token
Field
Field(int, char)
points int
color char
token Token
getToken() Token
collectToken() Token
getColor() char
setToken(Token) void
getPoints() int
Token
action(Player) void
isCollectable() boolean
getSymbol() char
getName() String
Game
Game(String[])
discardingPileBlue DiscardPile
discardingPilePurple DiscardPile
drawingPile DrawingPile
tokens ArrayList<Token>
FIELDS Field[]
discardingPileOrange DiscardPile
discardingPileGreen DiscardPile
discardingPileRed DiscardPile
players ArrayList<Player>
createDecks() void
getDiscardingPileBlue() DiscardPile
allFiguresOfOneInFinishArea() boolean
getDiscardingPileGreen() DiscardPile
gameOver() boolean
getPlayerAmount() int
canDrawFromDiscarding(Card) boolean
getPlayers() ArrayList<Player>
getDiscardingPileOrange() DiscardPile
getDiscardingPileRed() DiscardPile
figuresInFinishArea() int
getDrawingPile() DrawingPile
placeTokens() void
getWinningPlayer() Player
getDiscardingPilePurple() DiscardPile
Player
Player(String, char, Game)
symbol char
playedCardsOrange DirectionDiscardPile
hand ArrayList<Card>
playedCardsPurple DirectionDiscardPile
playedCardsBlue DirectionDiscardPile
game Game
lastTrashed Card
name String
lastMovedFigure Figure
figures Figure[]
playedCardsRed DirectionDiscardPile
victoryPoints int
playedCardsGreen DirectionDiscardPile
tokens ArrayList<Token>
goblinSpecialPlayed boolean
allFiguresGoblin() boolean
getSymbol() char
sortColorList(ArrayList<Card>) ArrayList<Card>
getName() String
canPlay() boolean
calcTokenPoints() void
setLastTrashed(Card) void
drawCardsUpToEight(DrawingPile) void
getSortedHand() ArrayList<Card>
getVictoryPoints() int
placeFigure(char, Figure) void
getFigureByPos(int) Figure
getPlayedCardsBlue() DirectionDiscardPile
getGame() Game
setName(String) void
cardFitsToAnyPile(Card) boolean
getHand() ArrayList<Card>
getFigureAmountOnField(int) int
isGoblinSpecialPlayed() boolean
getPlayedCardsGreen() DirectionDiscardPile
getPlayedCardsPurple() DirectionDiscardPile
putCardOnDiscardingPile(Card) void
getLastTrashed() Card
getLastMovedFigure() Figure
getTokens() ArrayList<Token>
drawFromDiscardingPile(DiscardPile) void
drawFromDrawingPile(DrawingPile) void
getPlayedCardsOrange() DirectionDiscardPile
addCardToPlayedCards(Card) void
playGoblinSpecial() void
getPlayedCardsRed() DirectionDiscardPile
getCardFromHand(String) Card
goblinSpecialPoints() int
getFigureAmountInFinishArea() int
getFigures() Figure[]
setVictoryPoints(int) void
Figure
Figure(char)
pos int
latestPos int
color char
move(char) void
setPos(int) void
getPos() int
getLatestPos() int
DirectionDiscardPile
DirectionDiscardPile(char)
direction int
cardFitsToPile(Card) boolean
add(Card) void
getDirection() int
DiscardPile
DiscardPile(char)
color char
pile ArrayList<Card>
isEmpty() boolean
getAndRemoveTop() Card
getPile() ArrayList<Card>
getTop() Card
add(Card) void
getColor() char
Card
Card(int, char)
Card(int, char, int)
number int
color char
oracleNumber int
getNumber() int
getColor() char
DrawingPile
DrawingPile()
cards Stack<Card>
draw() Card
isEmpty() boolean
getCards() Stack<Card>
Controller
Controller()
model Game
view View
waitForConfirmation(String) void
setView(View) void
playCard(Player) void
doTokenSpiral(Player) void
doTokenGoblin(Player) void
startGame() void
trashCard(Player) void
takeTurn(Player) void
drawOne(Player) void
usingToken(Player) void
Cli
Cli()
inputPlayersNames(int) String[]
printHand(Player) void
printBoardPart(int, int, ArrayList<Player>) void
promptCardString(String) String
printTopCards(Player) void
printDirection(DirectionDiscardPile) String
promptInt(int, int, String) int
promptPlayersChoice(String) boolean
printTop(DirectionDiscardPile) void
error(String) void
printCurrentBoard(Game) void
promptColor(String) char
printResults(Game) void
printDiscardingPiles(Game) void
out(String) void
printTop(DiscardPile) void
promptColorAndHand(String) char
View
promptPlayersChoice(String) boolean
promptColor(String) char
printCurrentBoard(Game) void
printHand(Player) void
inputPlayersNames(int) String[]
printTopCards(Player) void
printDiscardingPiles(Game) void
promptInt(int, int, String) int
promptColorAndHand(String) char
printResults(Game) void
error(String) void
promptCardString(String) String
out(String) void
Main
Main()
main(String[]) void
«create»
model
view
game
players
figures
playedCardsRed
discardingPileRed
drawingPile
tokens
tokenTwo
token
FIELDS
card
pile
hand

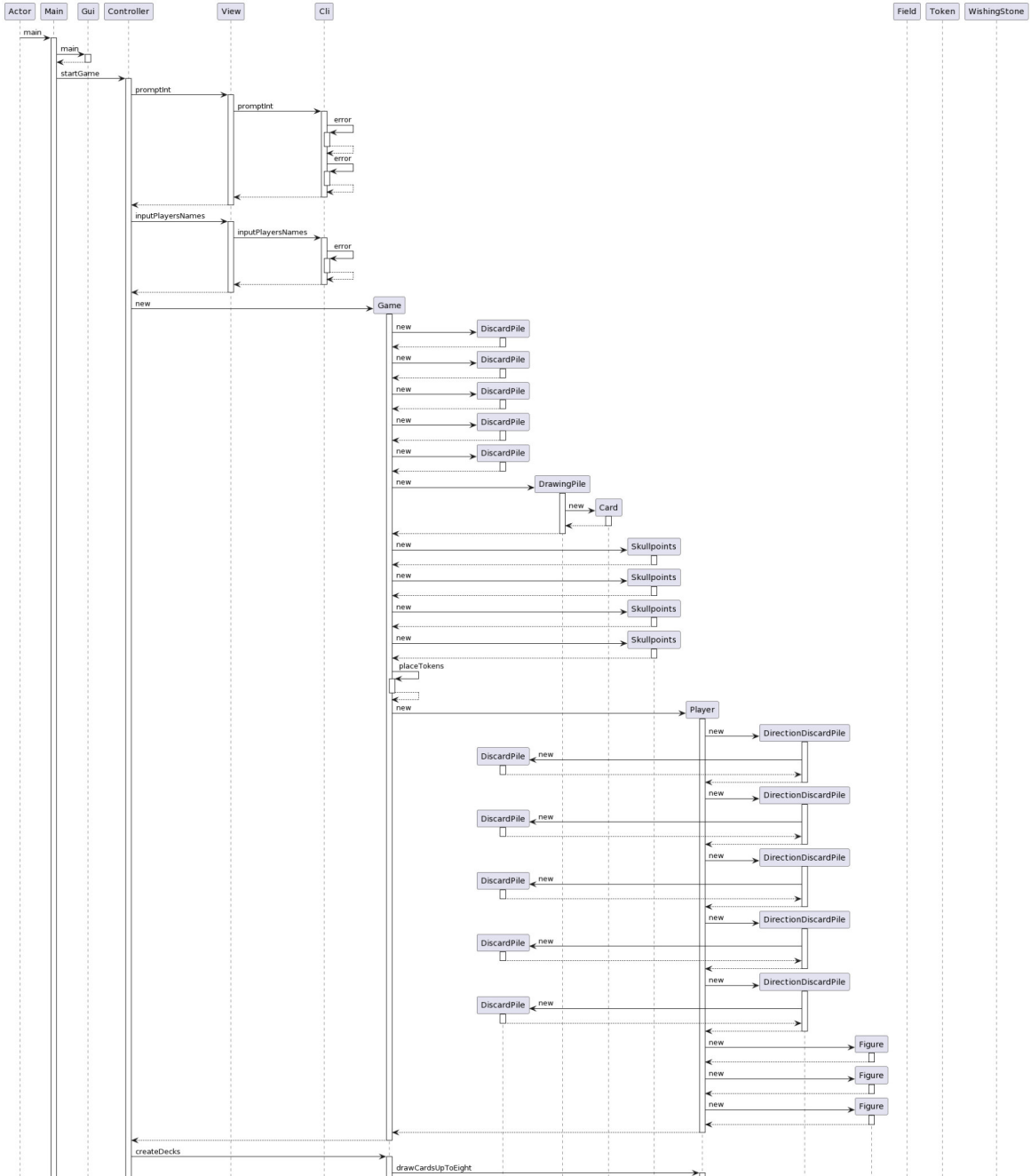
ZUSTANDSDIAGRAMM

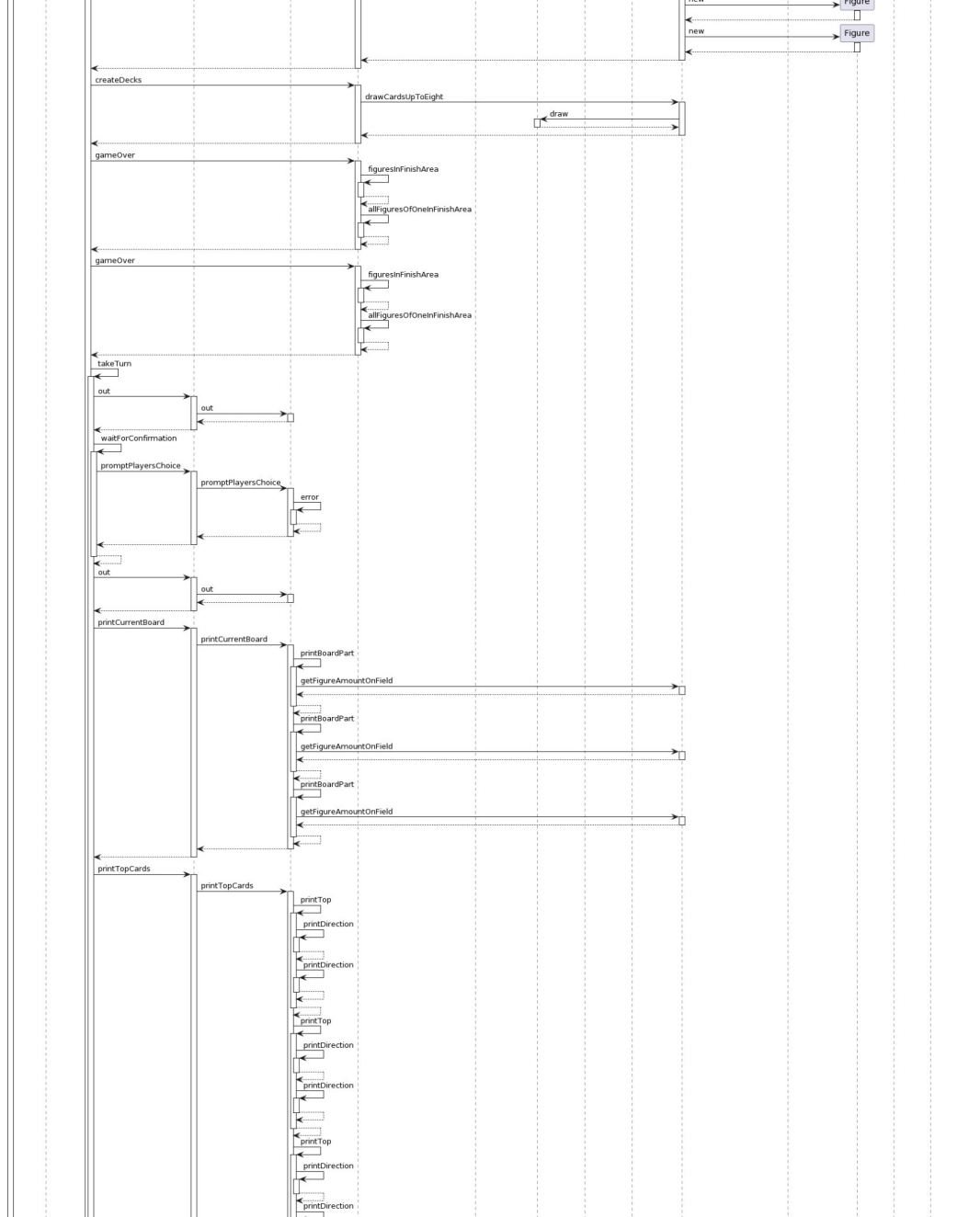
Gruppe 5


GameOn
preparation
entry choose player amount
choose bots or human players
type names[human players chosen]
fields are created[player amount + bots/humans chosen, names typed]
start of player's turn
place card[fits on pile]
player places card
entry choose if figure or oracle is placed
exit play token if exists on field
move figure[chose figure]
move oracle forwards[chose oracle and step amount]
get oracle bonus[moved oracle to field with at least 1 figure]
[token on field]
found token
entry execute token-action
[token-action done]
[finished turn]
next player's turn[has 8 cards on hand]
draw Card(s)
entry choose if you want to draw from discarding or drawing pile
draw from discarding pile[chose discarding pile]
draw from drawing pile[chose drawing pile]
[last card has been drawn from drawing pile]
[goal area has 3 figures from one player or 5 figures in total]
GameOver
entry calculate points

SEQUENZDIAGRAMM

Gruppe 5


Actor
Main
Gui
Controller
View
Cli
Field
Token
WishingStone
main
main
startGame
promptInt
promptInt
error
error
inputPlayersNames
inputPlayersNames
error
new
Game
new
DiscardPile
new
DiscardPile
new
DiscardPile
new
DiscardPile
new
DiscardPile
new
DrawingPile
new
Card
new
Skullpoints
new
Skullpoints
new
Skullpoints
new
Skullpoints
placeTokens
new
Player
new
DirectionDiscardPile
DiscardPile
new
new
DirectionDiscardPile
DiscardPile
new
new
DirectionDiscardPile
DiscardPile
new
new
DirectionDiscardPile
DiscardPile
new
new
DirectionDiscardPile
DiscardPile
new
new
Figure
new
Figure
new
Figure
createDecks
drawCardsUpToEight


new
Figure
new
Figure
createDecks
drawCardsUpToEight
draw
gameOver
figuresInFinishArea
allFiguresOfOneInFinishArea
gameOver
figuresInFinishArea
allFiguresOfOneInFinishArea
takeTurn
out
out
waitForConfirmation
promptPlayersChoice
promptPlayersChoice
error
out
out
printCurrentBoard
printCurrentBoard
printBoardPart
getFigureAmountOnField
printBoardPart
getFigureAmountOnField
printBoardPart
getFigureAmountOnField
printTopCards
printTopCards
printTop
printDirection
printDirection
printTop
printDirection
printDirection
printTop
printDirection
printDirection