Final flow

Variables:
- numberOfPlayers (1-4) - from input (Scanner???)

Methods:
- countnumberOfPlayers

Input numbers of...

Variables:
- playerName + numberOfRugs (default 15) + numberOfDirhmas (default 30) :
  - player1: rug0 + numberOfDirhmas1
  - player2: rug0 + numberOfDirhmas2
  - player3: rug0 + numberOfDirhmas3
  - player4: rug0 + numberOfDirhmas4
  - → game order auto go from 1 to 4

Methods:
- createPlayers
  - assignRugs
  - createBagOfDirhams

Assign of order, r...

Variables:
- assamPosition (default at (4,4) - 0)
- boardMatrix[][] [7×7]

Game starts,...

Variables:
- assamPosition (default at (4,4) - 0)

Methods:
- rotationAssam(): 0, 90, 180, 270 (clockwise) → return assamPosition (0,4) - r)

1st playing turn

Rotation Assam

Variables:
- diceValue[]: 1, 2, 2, 3, 3, 4
- diceNumber

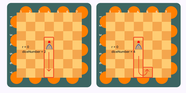
Methods:
- randomDice(): choose random value in diceValue[] → return int diceNumber

Dice roll

Variables:
- assamPosition (0,4) - r)
- diceNumber

Methods:
- transformAssam(): = x + y
→ return assamPosition (0,4 + x,4 +y) - r)
- assamwithinBoard(): check if Assam are inside board
- assamReachEdge(): rotate Assam
  - continue move rest of squares

Move Assam



Variables:
- rugPosition { (x, y) - r}

Methods:
- rugPossibleRegion(): the possible area that the rug can be placed, around Assam
- placeRug()
- rugwithinBoard(): check if rug are inside board

Place rug

Variables:
- playerName
- numberOfRugs
- numberOfDirhmas

Methods:
- playerString(Updates): update player string with new numberOfRugs and numberOfDirhmas

Player String upd...

Repeat

Variables:
- assamPosition ((x,y) - r)

Methods:
- rotationAssam(): 0, 90, 180, 270 (clockwise) → return assamPosition (0,4) - r)

next playing turn...

Rotation Assam

Variables:
- diceValue[]: 1, 2, 2, 3, 3, 4

Methods:
- randomDice(): choose random value in diceValue[] → return int

Dice roll

Variables:
- assamPosition ((x,y) - r)

Methods:
- transformAssam(): = x + y
→ return assamPosition (0,y) - r)
- assamwithinBoard(): check if Assam are inside board
- assamReachEdge(): rotate Assam
  - continue move rest of squares

Move Assam

Variables:
- assamPosition ((x,y) - r)
- rugPosition { (x, y) - r}
- int rugRegion
- numberOfDirhmas (1,2,3,4)

Methods:
- assamLandsRug(): compare if Assam's index is the same with any rugs' index
- rugRegion(): count connected region of squares that are covered by rugs of the same colour, except the one Assam is landing on
→ return int rugRegion
- rugConnected(): check if rugs connected/sharing an edge
- payDirhams(): take amount of dirhams (= rugRegion) from amountofDirhmas of this player to another
- calDirhmas(): calculate amount of dirhams (= rugRegion)
- outOfDirhams(): check if player out of money

Paid money

Variables:
- rugPosition { (x, y) - r}

Methods:
- placeRug()
- rugwithinBoard(): check if rug are inside board
- rugsOverlap(): check if cover both squares of another rug (must be all visible) that is already on the board
- outOfRug(): check if player out of rugs

Place rug

Variables:
- playerName
- numberOfRugs
- numberOfDirhmas

Methods:
- playerString(Updates): update player string with new numberOfRugs and numberOfDirhmas

Player String upd...

Variables:
- numberOfDirhmas
- numberOfRugs
- playerScore

Methods:
- playerOut(): check if player's numberOfRugs = 0 or not, then count numberOfDirhmas
→ note player's scores
- calScore(): calculate player's score
→ note player's scores

Players lose (out)

Count dirhams

Variables:
- amountofDirhmas

Methods:
- calDirhmas()

Out of money

Out of rugs

Variables:
- numberRugs

Methods:
- outofRugs: check if numberRugs = 0

Variables:
- numberOfDirhmas
- numberOfRugs
- playerScore

Methods:
- playerOut(): check if player's numberOfRugs = 0 or not, then count numberOfDirhmas
→ note player's scores
- calScore(): calculate player's score
→ note player's scores

Last player out

Count dirhams

Variables:
- amountofDirhmas

Methods:
- calDirhmas()

Out of rugs

Variables:
- numberRugs

Methods:
- outofRugs: check if numberRugs = 0

Find winner

Compare scores

Variables:
- playerScore

Methods:
- compareScores(): compare all players' score, no equal scores
→ return player win

Equal sc...

Variables:
- playerScore

Methods:
- compareScores(): compare all players' score, there are equal scores

No

Yes

2 equal s...

Variables:
- playerScore

Methods:
- compareScores(): compare all players' score, there are 2 equal scores

Compare...

Variables:
- amountofDirhmas

Methods:
- compareDirhmas(): compare to find bigger amount of amountofDirhmas

> 2 equal...

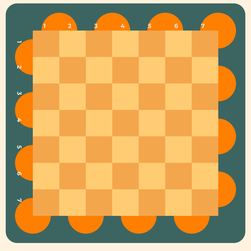
Variables:
- playerScore

Methods:
- compareScores(): compare all players' score, there are more than 2 equal scores
→ return game drawn

GAME DRAW

Methods:
- gameResult(): show game result (game drawn)

Methods:
- gameResult(): show game result (winner + players' score from high to low)

GAME ENDS

Design - elements