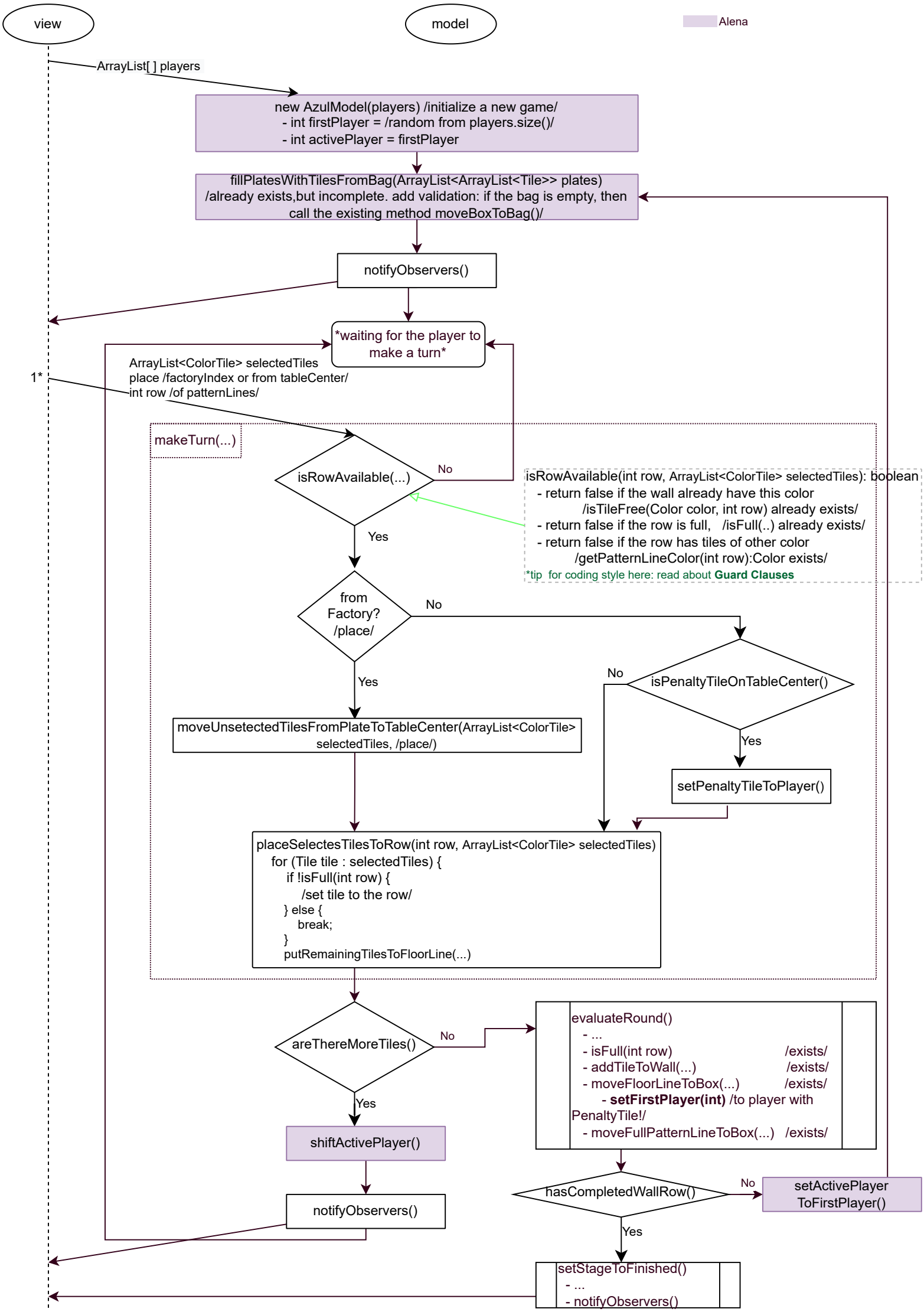
view
model
Alena
ArrayList[ ] players
new AzulModel(players) /initialize a new game/
- int firstPlayer = /random from players.size()/
- int activePlayer = firstPlayer
fillPlatesWithTilesFromBag(ArrayList<ArrayList<Tile>> plates)
/already exists,but incomplete. add validation: if the bag is empty, then call the existing method moveBoxToBag()/
notifyObservers()
waiting for the player to make a turn
1*
ArrayList<ColorTile> selectedTiles
place /factoryIndex or from tableCenter/
int row /of patternLines/
makeTurn(...)
isRowAvailable(...)
No
Yes
isRowAvailable(int row, ArrayList<ColorTile> selectedTiles): boolean
- return false if the wall already have this color
/isTileFree(Color color, int row) already exists/
- return false if the row is full, /isFull(..) already exists/
- return false if the row has tiles of other color
/getPatternLineColor(int row):Color exists/
*tip for coding style here: read about Guard Clauses
from Factory? /place/
No
Yes
isPenaltyTileOnTableCenter()
No
Yes
setPenaltyTileToPlayer()
moveUnsetectedTilesFromPlateToTableCenter(ArrayList<ColorTile> selectedTiles, /place/)
placeSelectesTilesToRow(int row, ArrayList<ColorTile> selectedTiles)
for (Tile tile : selectedTiles) {
if !isFull(int row) {
/set tile to the row/
} else {
break;
}
putRemainingTilesToFloorLine(...)
areThereMoreTiles()
No
Yes
evaluateRound()
- ...
- isFull(int row) /exists/
- addTileToWall(...) /exists/
- moveFloorLineToBox(...) /exists/
- setFirstPlayer(int) /to player with PenaltyTile!/
- moveFullPatternLineToBox(...) /exists/
shiftActivePlayer()
notifyObservers()
hasCompletedWallRow()
No
setActivePlayer ToFirstPlayer()
Yes
setStageToFinished()
- ...
- notifyObservers()

1* selectedTiles not empty AND click on a row from patternLines (from activePlayer !!!)