ADT Board

Service : **Board**
Type : int, bool
Require : Block, Grid

Observators :
getCurrentBlock : [Board] -> Block
isCleaned : [Board] − > bool
getBrid : [Board] -> Grid
getNbLastCleaned : [Board] -> int
canRotateLeft : [Board] -> bool
précondition : *canRotateLeft*(B) require *isBlock*(B)
canRotateRight : [Board] -> bool
précondition : *canRotateRight*(B) require *isBlock*(B)
canGoLeft : [Board] -> bool
précondition : *canGoLeft*(B) require *isBlock*(B)
cangoRight : [Board] -> bool
précondition : *cangoRight*(B) require *isBlock*(B)
isBlock : [Board] -> bool
isBottom : [Board] -> bool
précondition : *isBottom*(B) require *isBlock*(B)
isConflict : [Board] -> bool
précondition : *isConflict*(B) require *isBlock*(B)
getXblock : [Board] * int -> int
précondition : *getXblock*(B,x) require x >= *getcurrentBlock*(*getXMin*(B))
&& x <= *getcurrentBlock*(*getXMax*(B))
getYblock : [Board] * int -> int
précondition : *getYblock*(B,y) require y >= *getcurrentBlock*(*getYMin*(B))
&& y <= *getcurrentBlock*(*getYMax*(B))
getXMinBlock : [Board] -> int
getYMinBlock : [Board] -> int
getBottomHeight : [Board] -> int
précondition : *getBottomHeight*(B) require *isBlock*(B)

Constructor :
init : int * int -> [Board]
précondition : *init*(x,y) require $x > 0$ && $y > 0$ && $y >= x$

Opérations :
doRotateLeft : [Board] -> [Board]
précondition : *doRotateLeft*(B) require *isBlock*(B) && *canRotateLeft*(B)
doLeft : [Board] -> [Board]
précondition : *doLeft*(B) require *isBlock*(B) && *cangoLeft*(B)

doRotateRight : [Board] -> [Board]
précondition : *doRotateRight*(B) require *isBlock*(B) && *canRotateRight*(B)
doRight : [Board] -> [Board]
précondition : *doRight*(B) require *isBlock*(B) && *cangoRight*(B)
doBottom : [Board] -> [Board]
précondition : *doBottom*(B) require *isBlock*(B)
step : [Board] -> [Board]
précondition : *step*(B) require *isBlock*(B)
insert : [Board] * Block -> [Board]
précondition : *insert*(B) require ¬*isBlock*(B)
remove : [Board] -> [Board]
précondition : *remove*(B) require *isBlock*(B)
clean : [Board] -> [Board]
précondition : *clean*(B) require *isBlock*(B) && *isBottom*(B)

Observations :

- *invariants*
  1<=*getXblock*(B,**Block :***getXMin*(*getcurrentBlock*(B))) &&
  getXblock(B,**Block :***getXMax*(*getcurrentBlock*(B)))<=**Grid :***getWidth*(*getgrid*(B))+1
  1<=*getYblock*(B,**Block :***getYMin*(*getcurrentBlock*(B))) &&
  getYblock(B,**Block :***getYMax*(*getcurrentBlock*(B)))<=**Grid :***getHeight*(*getgrid*(B))+4
  getNbLastCleaned(B) >= 0
  isBlock(B) <=> *getXMinBlock*(B) >= 0 && *getXMinBlock*(B) <= *getgrid*(*getWidth*(B)))
  isBlock(B) <=> *getYMinBlock*(B) >= 0 && *getYMinBlock*(B) <= *getgrid*(*getHeight*(B)))+1

- *init*
  getgrid(B) = **Grid :***init*(x,y)
  isBlock(*init*(x,y)) = false
  Grid :*getWidth*(*getgrid*(*init*(x,y))) = x
  Grid :*getHeight*(*getgrid*(*init*(x,y))) = y
  getNbLastCleaned(*init*(*x*, *y*)) = 0
  getXMinBlock(*init*(*x*, *y*)) = 0
  getYMinBlock(*init*(*x*, *y*)) = 0

- *doRotateLeft*
  getNbLastCleaned(*doRotateLeft*(B)) = *getNbLastCleaned*(B)
  isBlock(*doRotateLeft*(B)) = *isBlock*(B)
  getXMinBlock(*doRotateLeft*(B)) = *getXMinBlock*(B)
  getYMinBlock(*doRotateLeft*(B)) = *getYMinBlock*(B)
  getcurrentBlock(*doRotateLeft*(B)) = **Block** :*rotateLeft*(*getcurrentBlock*(B))

- *doLeft*
  getcurrentBlock(*getSize*(*doLeft*(B))) = *getcurrentBlock*(*getSize*(B))
  getcurrentBlock(*getType*(*doLeft*(B)) = *getcurrentBlock*(*getType(B))*
  getcurrentBlock(getNbPos((*doLeft*(B))) = *getcurrentBlock*(*getNbPos*(B))
  getNbLastCleaned(*doLeft*(B)) = *getNbLastCleaned*(B)
  isBlock(*doLeft*(B)) = *isBlock*(B)
  getXMinBlock(*doLeft*(B)) = *getXMinBlock*(B)-1
  getYMinBlock(*doLeft*(B)) = *getYMinBlock*(B)

- *doRotateRight*
  getNbLastCleaned(*v*(B)) = *getNbLastCleaned*(B)
  isBlock(*doRotateRight*(B)) = *isBlock*(B)
  getXMinBlock(*doRotateRight*(B)) = *getXMinBlock*(B)
  getYMinBlock(*doRotateRight*(B)) = *getYMinBlock*(B)
  getcurrentBlock(*doRotateRight*(B)) = **Block** :*rotateRight*(*getcurrentBlock*(B))

- *doRight*
  getcurrentBlock(*getSize*(*doRight*(B))) = *getcurrentBlock*(*getSize*(B))
  getcurrentSize(*getType*(*doRight*(B))) = *getcurrentBlock*(*getType*(B))
  getcurrentNbPos(*getNbPos*(*doRight*(B))) = *getcurrentBlock*(*getNbPos*(B))
  getNbLastCleaned(*doRight*(B)) = *getNbLastCleaned*(B)
  isBlock(*doRight*(B)) = *isBlock*(B)
  getXMinBlock(*doRight*(B)) = *getXMinBlock*(B)+1
  getYMinBlock(*doRight*(B)) = *getYMinBlock*(B)

- *doBottom*
  getcurrentBlock(*getSize*(*doBottom*(B))) = *getcurrentBlock*(*getSize*(B))
  getcurrentBlock(*getType*(*doBottom*(B))) = *getcurrentBlock*(*getType*(B))
  getcurrentBlock(*getNbPos*(*doBottom*(B))) = *getcurrentBlock*(*getNbPos*(B))
  isBlock(*doBottom*(B)) = *isBlock*(B)
  getXMinBlock(*doBottom*(B)) = *getXMinBlock*(B)
  getYMinBlock(*doBottom*(B)) = *getYMinBlock*(B) + *getBottomHeight*(B) + *getN-bLastCleaned*(B)
  getBottomHeight(*doBottom*(B)) - *getNbLastCleaned*(B) = 0 || *isBottom*(B)

- *step*
  ¬*isBottom*(B) <=> *isBlock*(*step*(B)) = *isBlock*(B)
  ¬*isBottom*(B) <=> *getXMinBlock*(*step*(B)) = *getXMinBlock*(B)
  ¬*isBottom*(B) <=> *getYMinBlock*(*step*(B)) = *getYMinBlock*(B) + 1
  ¬*isBottom*(B) <=> *getBottomHeight*(*step*(B)) = *getBottomHeight*(B) - 1

- *insert*
  getCurrentBlock(*insert*(B, block)) = block
  isBlock(*insert*(B, block)) = true
  getSize(block,*insert*(B))) = *getcurrentBlock*(*getSize*(B))
  getType(block,*insert*(B))) = *getcurrentBlock*(*getType*(B))
  getNbPos(block,*insert*(B))) = *getcurrentBlock*(*getNbPos*(B))

- *remove*
  getcurrentBlock(*remove*(B)) = null
  ∀ x (1 <= x && x <= *getgrid*(*getWidth*(B))){
  ∀ y (1 <= y && y <= *getgrid*(*getHeight*(B))){
  getgrid(*isOccupied*(*remove*(B),x,y)) = *getgrid*(*isOccupied*(B,x,y))
  }
  getNbLastCleaned(*remove*(B)) = *getNbLastCleaned*(B)
  isBlock(*remove*(B)) = false
  getXMinBlock(*remove*(B)) = 0
  getYMinBlock(*remove*(B)) = 0

- *clean*
  isBlock(*remove*(B)) = false
  isBottom(*remove*(B)) = false
  getcurrentBlock(*getSize*(*insert*(B))) = *getcurrentBlock*(*getSize*(B))
  getcurrentBlock(*getType*(*insert*(B))) = *getcurrentBlock*(*getType*(B))
  getcurrentBlock(*getNbPos*(*insert*(B))) = *getcurrentBlock*(*getNbPos*(B))