

ADT Joueur

Service : **Joueur**

Type : int, bool

Require : Tetris

Observers :

getTetris : [Joueur] -> Tetris

canPlay : [Joueur] -> bool

Constructor :

init : -> [Joueur]

Opérations :

startGame : [Joueur] -> [Joueur]

précondition : *startGame*(J) require $\neg canPlay(J)$

goLeft : [Joueur] -> [Joueur]

précondition : *goLeft*(J) require *canPlay*(J)

goRight : [Joueur] -> [Joueur]

précondition : *goRight*(J) require *canPlay*(J)

goDown : [Joueur] -> [Joueur]

précondition : *goDown*(J) require *canPlay*(J)

rotateLeft : [Joueur] -> [Joueur]

précondition : *rotateLeft*(J) require *canPlay*(J)

rotateRight : [Joueur] -> [Joueur]

précondition : *rotateRight*(J) require *canPlay*(J)

Observations :

- *invariants*
 $getTetris(J) \neq null$
- *init*
- *startGame*
 $getTetris(startGame(J)) = getTetris(J)$
 $getTetris(startGame(J)) = \mathbf{Tetris} : next(getTetris(J))$
- *goLeft*
 $getTetris(goLeft(J)) = getTetris(J)$
 $getTetris(goLeft(J)) = \mathbf{Tetris} : goLeft(getTetris(J))$

- *goRight*
 $getTetris(goRight(J)) = getTetris(J)$
 $getTetris(goRight(J)) = \mathbf{Tetris} : goRight(getTetris(J))$
- *goDown*
 $getTetris(goDown(J)) = getTetris(J)$
 $getTetris(goDown(J)) = \mathbf{Tetris} : goDown(getTetris(J))$
- *rotateLeft*
 $getTetris(rotateLeft(J)) = getTetris(J)$
 $getTetris(rotateLeft(J)) = \mathbf{Tetris} : rotateLeft(getTetris(J))$
- *rotateRight*
 $getTetris(rotateRight(J)) = getTetris(J)$
 $getTetris(rotateRight(J)) = \mathbf{Tetris} : rotateRight(getTetris(J))$