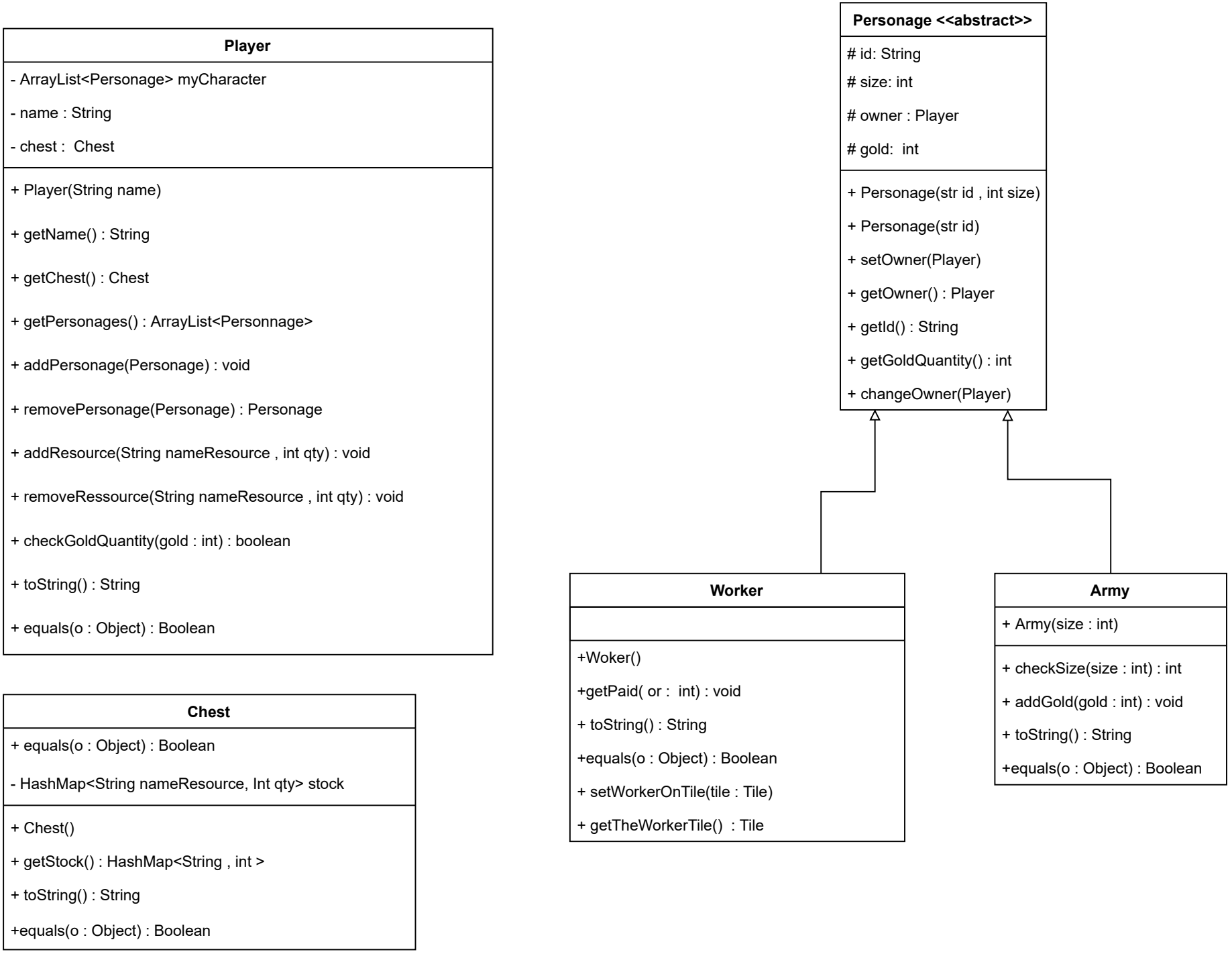
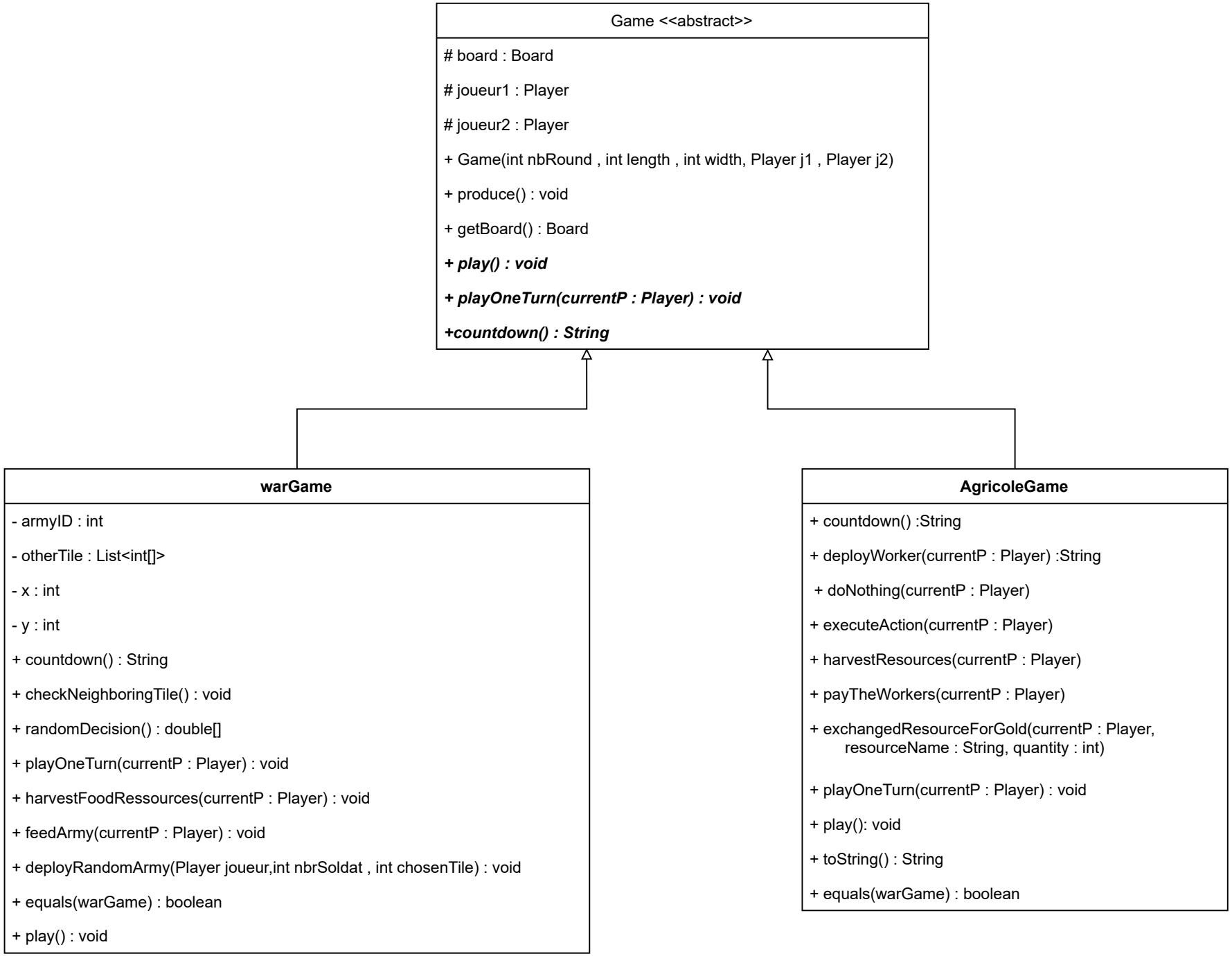
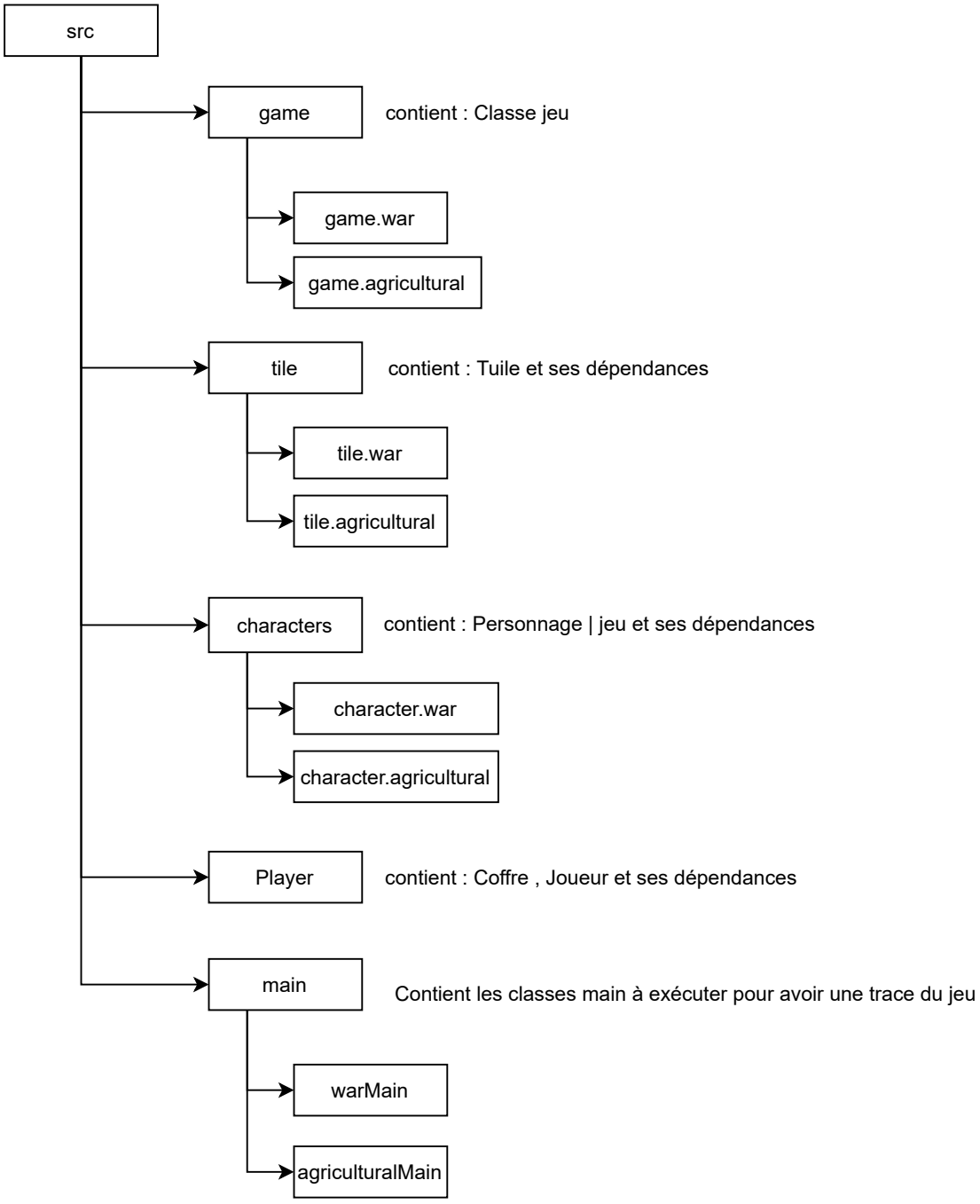
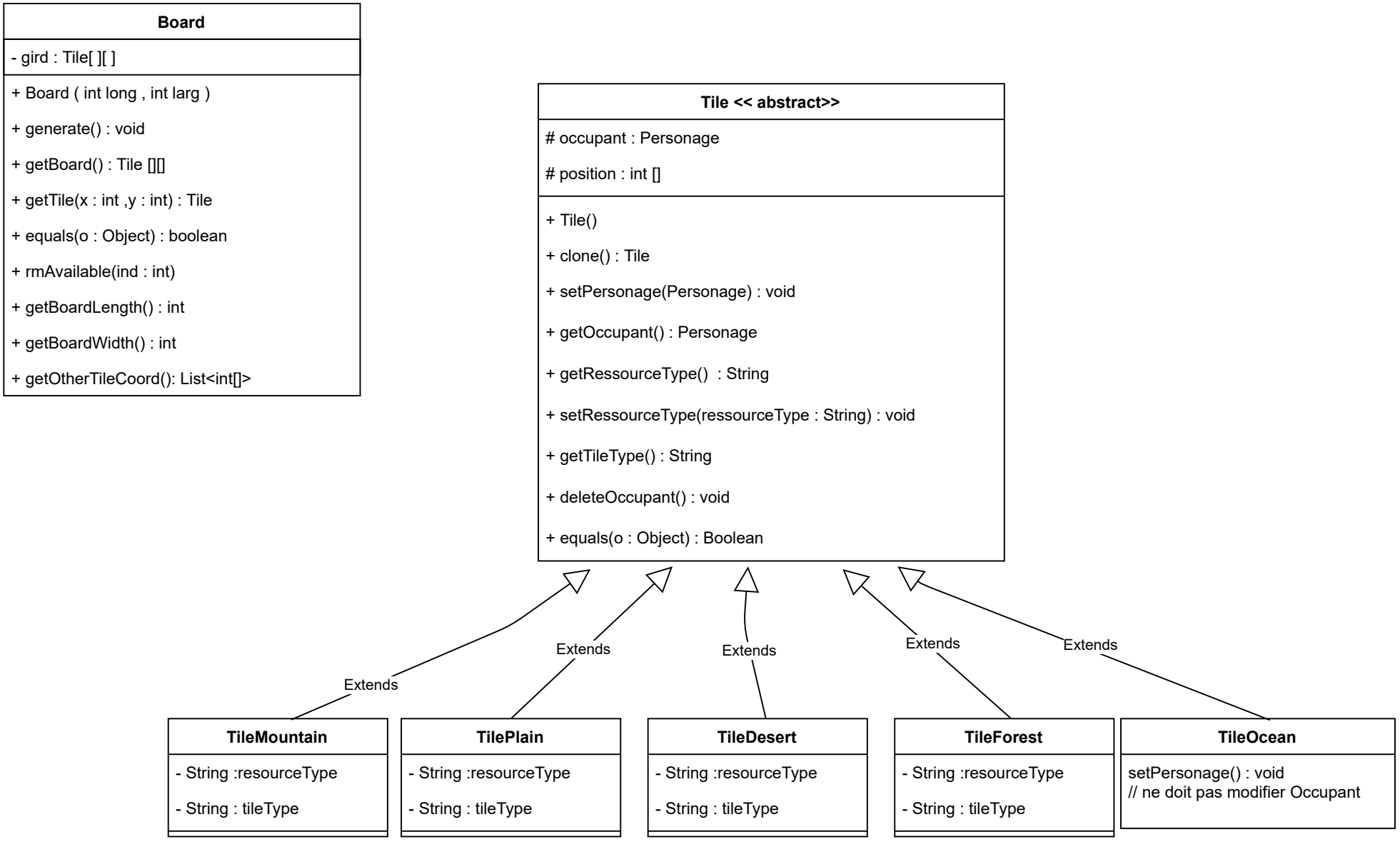
Diagramme UML du projet




Game <<abstract>>
board : Board
joueur1 : Player
joueur2 : Player
+ Game(int nbRound , int length , int width, Player j1 , Player j2)
+ produce() : void
+ getBoard() : Board
+ play() : void
+ playOneTurn(currentP : Player) : void
+countdown() : String
warGame
- armyID : int
- otherTile : List<int[]>
- x : int
- y : int
+ countdown() : String
+ checkNeighboringTile() : void
+ randomDecision() : double[]
+ playOneTurn(currentP : Player) : void
+ harvestFoodRessources(currentP : Player) : void
+ feedArmy(currentP : Player) : void
+ deployRandomArmy(Player joueur,int nbrSoldat , int chosenTile) : void
+ equals(warGame) : boolean
+ play() : void
AgricoleGame
+ countdown() :String
+ deployWorker(currentP : Player) :String
+ doNothing(currentP : Player)
+ executeAction(currentP : Player)
+ harvestResources(currentP : Player)
+ payTheWorkers(currentP : Player)
+ exchangedResourceForGold(currentP : Player,
resourceName : String, quantity : int)
+ playOneTurn(currentP : Player) : void
+ play(): void
+ toString() : String
+ equals(warGame) : boolean


src
game
contient : Classe jeu
game.war
game.agricultural
tile
contient : Tuile et ses dépendances
tile.war
tile.agricultural
characters
contient : Personnage | jeu et ses dépendances
character.war
character.agricultural
Player
contient : Coffre , Joueur et ses dépendances
main
Contient les classes main à exécuter pour avoir une trace du jeu
warMain
agriculturalMain


Board
- gird : Tile[ ][ ]
+ Board (int long , int larg)
+ generate() : void
+ getBoard() : Tile [][]
+ getTile(x : int ,y : int) : Tile
+ equals(o : Object) : boolean
+ rmAvailable(ind : int)
+ getBoardLength() : int
+ getBoardWidth() : int
+ getOtherTileCoord(): List<int[]>
Tile << abstract>>
occupant : Personage
position : int []
+ Tile()
+ clone() : Tile
+ setPersonage(Personage) : void
+ getOccupant() : Personage
+ getRessourceType() : String
+ setRessourceType(ressourceType : String) : void
+ getTileType() : String
+ deleteOccupant() : void
+ equals(o : Object) : Boolean
Extends
Extends
Extends
Extends
Extends
TileMountain
- String :resourceType
- String : tileType
TilePlain
- String :resourceType
- String : tileType
TileDesert
- String :resourceType
- String : tileType
TileForest
- String :resourceType
- String : tileType
TileOcean
setPersonage() : void
// ne doit pas modifier Occupant