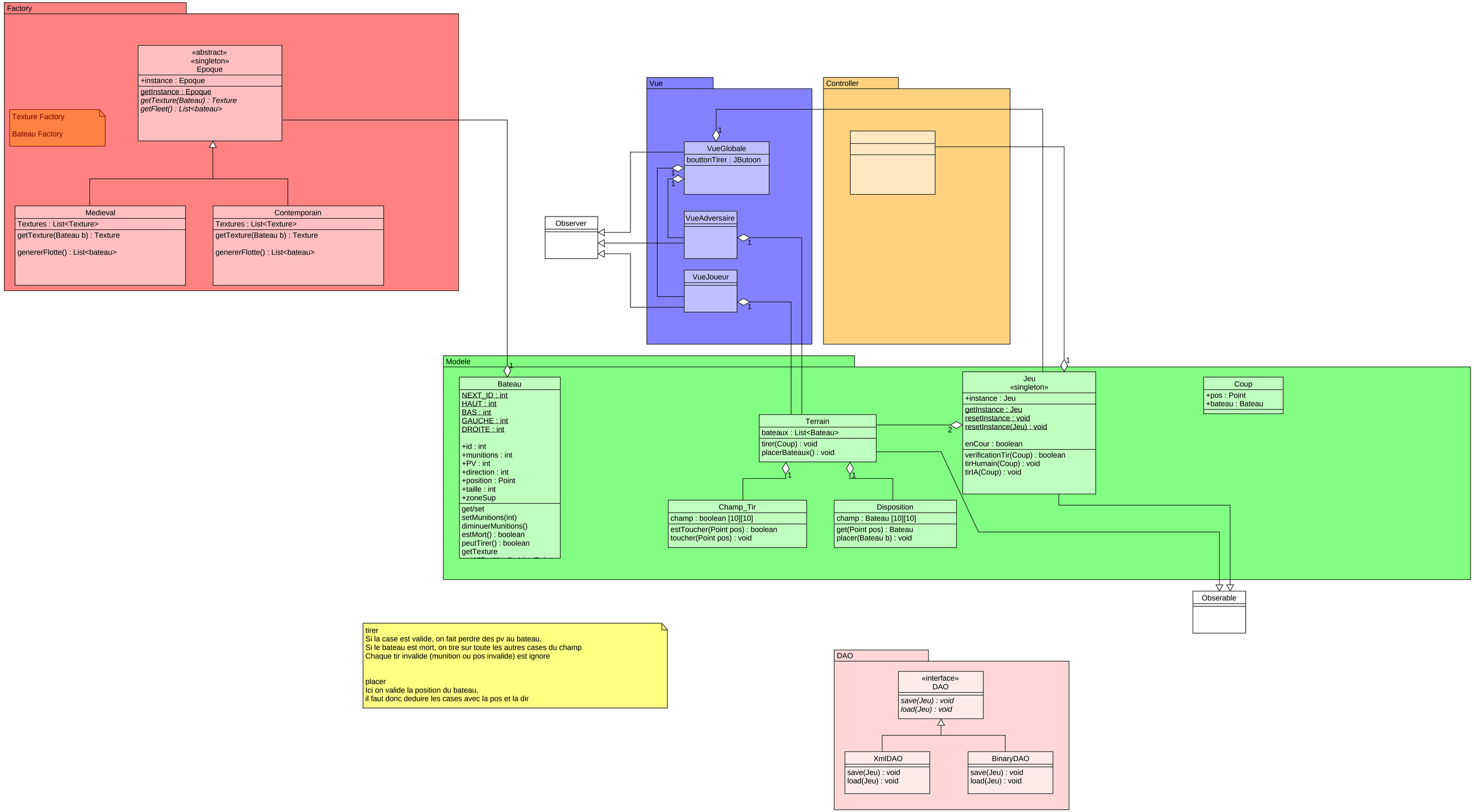
Factory
«abstract»
«singleton»
Epoque
+instance : Epoque
getInstance : Epoque
getTexture(Bateau) : Texture
getFleet() : List<bateau>
Texture Factory
Bateau Factory
Medieval
Textures : List<Texture>
getTexture(Bateau b) : Texture
genererFlotte() : List<bateau>
Contemporain
Textures : List<Texture>
getTexture(Bateau b) : Texture
genererFlotte() : List<bateau>
Vue
VueGlobale
bouttonTirer : JButoon
VueAdversaire
VueJoueur
Observer
Controller
Modele
Bateau
NEXT_ID : int
HAUT : int
BAS : int
GAUCHE : int
DROITE : int
+id : int
+munitions : int
+PV : int
+direction : int
+position : Point
+taille : int
+zoneSup
get/set
setMunitions(int)
diminuerMunitions()
estMort() : boolean
peutTirer() : boolean
getTexture
Terrain
bateaux : List<Bateau>
tirer(Coup) : void
placerBateaux() : void
Champ_Tir
champ : boolean [10][10]
estToucher(Point pos) : boolean
toucher(Point pos) : void
Disposition
champ : Bateau [10][10]
get(Point pos) : Bateau
placer(Bateau b) : void
Jeu
«singleton»
+instance : Jeu
getInstance : Jeu
resetInstance : void
resetInstance(Jeu) : void
enCour : boolean
verificationTir(Coup) : boolean
tirHumain(Coup) : void
tirIA(Coup) : void
Coup
+pos : Point
+bateau : Bateau
Observable
tirer
Si la case est valide, on fait perdre des pv au bateau,
Si le bateau est mort, on tire sur toute les autres cases du champ
Chaque tir invalide (munition ou pos invalide) est ignore
placer
Ici on valide la position du bateau,
il faut donc deduire les cases avec la pos et la dir
DAO
«interface»
DAO
save(Jeu) : void
load(Jeu) : void
XmlDAO
save(Jeu) : void
load(Jeu) : void
BinaryDAO
save(Jeu) : void
load(Jeu) : void