MAGE



WARRIOR





Modèle & API - TP1 [INF7845]

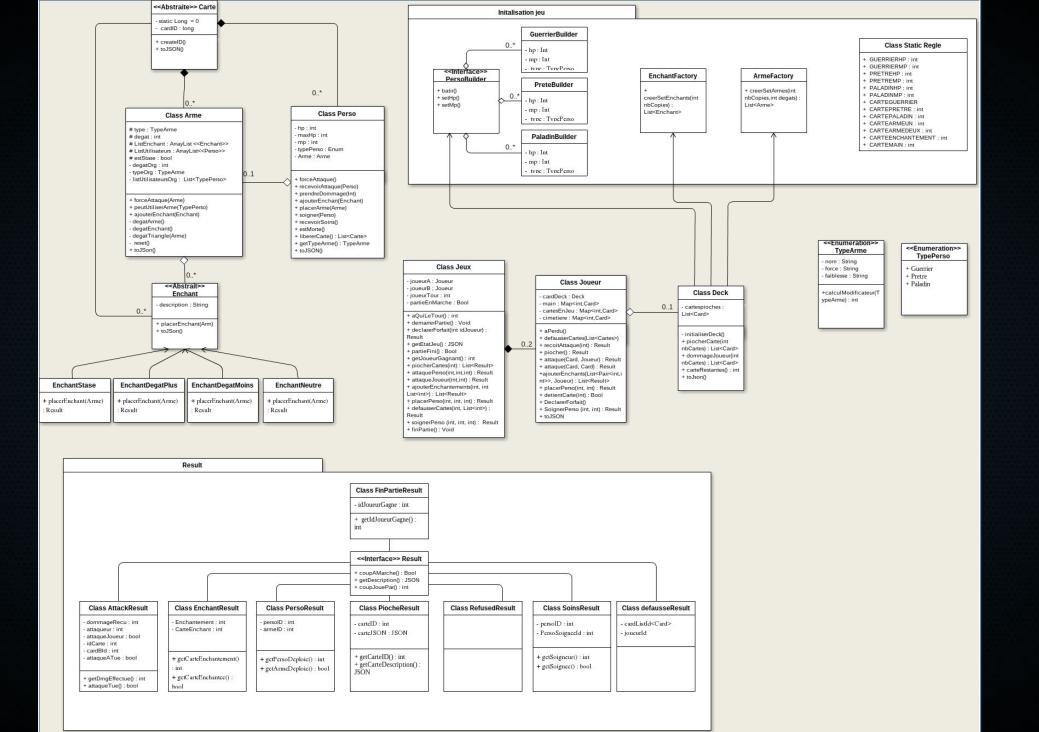
Partie II: Le retour du modèle

PALADIN

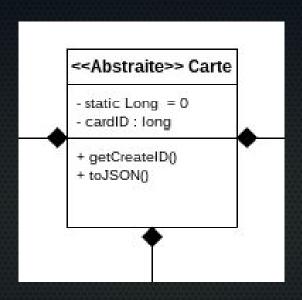


Nicolas Reynaud

Mathieu Gravel



I - Cartes



II - Cible



0..*

<<Abstrait>> Class Perso

- hp: int - maxHp: int - mp: int - maxMp: int - armePerso: Arme
- armeUtilisable : List<TypeArme>
- + forceAttaque() : int + forceAttaque(Perso) : int
- + prendreDegat(int, int) : AttaquePersoResult
- + libererCartes() : List<Carte> + ajouterEnchan(Enchant) : void + equiperArme(Arme) : boolean
- # recevoirSoins() : void
- + getMp() : int # utiliserMagie() : void + estMorte() : boolean
- + getArmesUtilisables() : List<TypeArme>
- + getTypeArme() : TypeArme
- + toJSON() : JSON
- + getArme() : Arme
- + peutEtreAttaquer() : boolean
- + recevoirAttaque(Perso) : AttaquePersoResult

<<Interface>> Cible

- + peutEtreAttaquer() : boolean
- + recoitAttaque(Perso) : Resultat
- + estMort() : boolean

Class Joueur

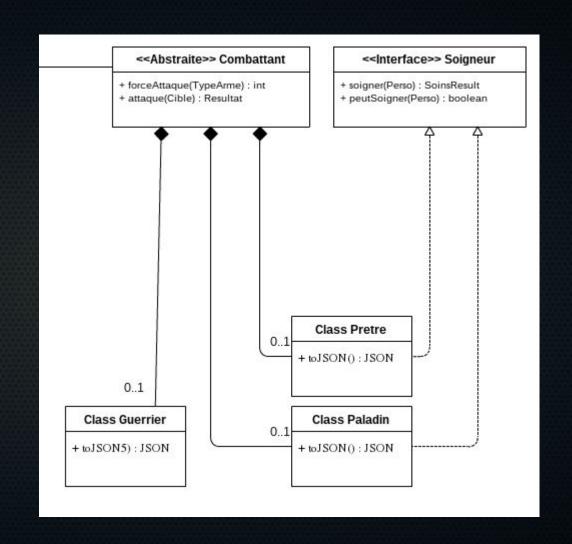
- idJoueur : int
- cardDeck : Deck
- main: Map<int,Card>
- cartesEnJeu : Map<int,Perso>
- cimetiere : List<int>
- + aPerdu()
- + defausserCartes(List<Cartes>)
- + recoitAttaque(int) : Result
- + piocher() : Result
- + attaque(Card, Joueur) : Result
- + attaque(Card, Card) : Result
- +ajouterEnchants(List<Pair<int,i nt>>, Joueur) : List<Result>
- + placerPerso(int, int) : Result
- + DeclarerForfait()
- + SoignerPerso (int, int) : Result
- + toJSON

III - Attaquant & Soin

<<Abstraite>> Combattant <<Interface>> Soigneur + forceAttaque(TypeArme) : int + attaque(Cible) : Resultat + soigner(Perso) : SoinsResult + peutSoigner(Perso) : boolean



IV - Les 3 types de perso



V - Les armes et leurs types

0..* Class Arme # type : TypeArme # degat : int # ListEnchant : ArrayList << Enchant>> # ListUtilisateurs : ArrayList<<Perso>> # estStase : bool - degatOrg : int - typeOrg : TypeArme 0..1 - listUtilisateursOrg : List<TypePerso> + forceAttaque(Arme) + peutUtiliserArme(TypePerso) + ajouterEnchant(Enchant) - degatArme() - degatEnchant() - degatTriangle(Arme) - reset() + toJSon()

Enum TypeArme

- Contondant : TypeArme

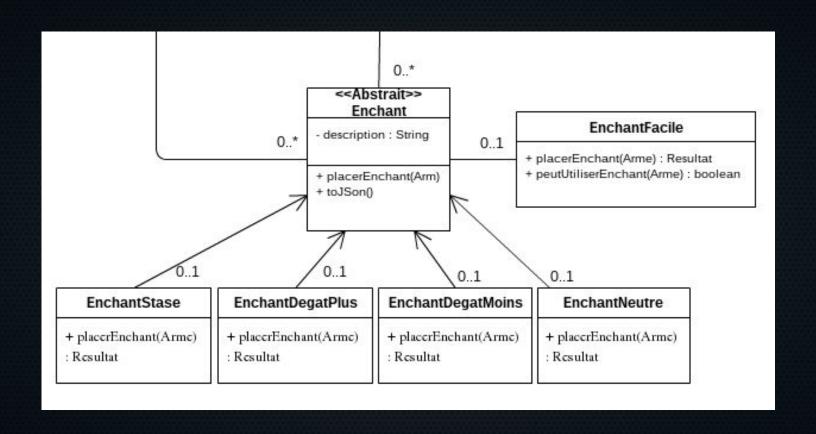
- Neutre : TypeArme

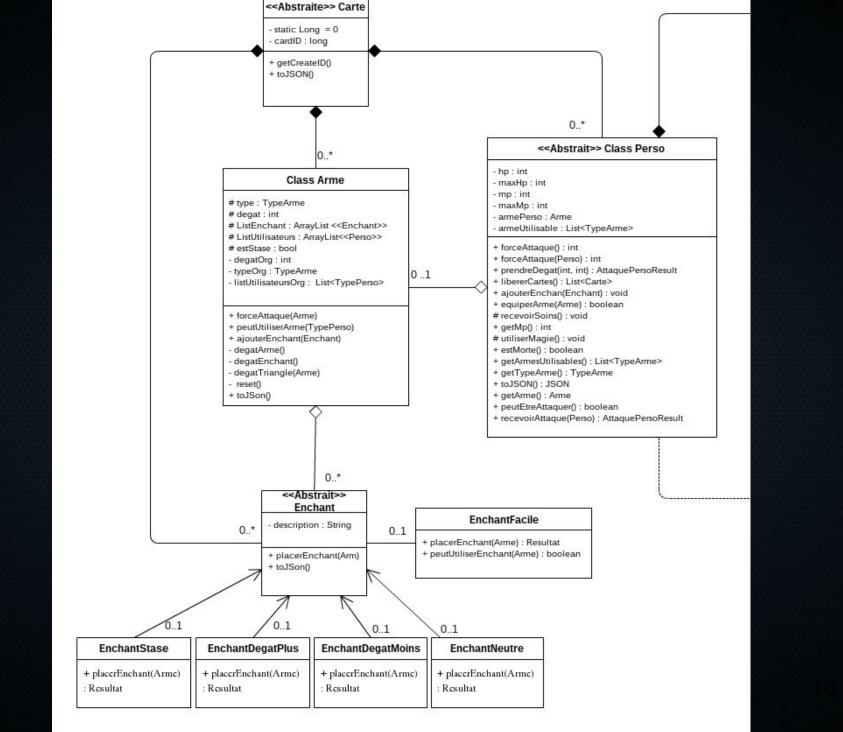
- Perforant : TypeArme

- Tranchant : TypeArme

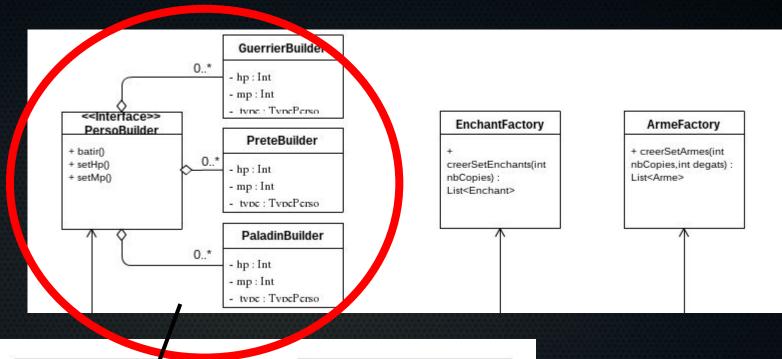
+ calculModificateur(TypeArme) : int

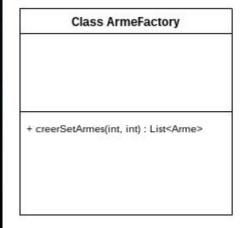
VI - Les enchantes

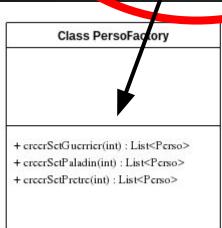


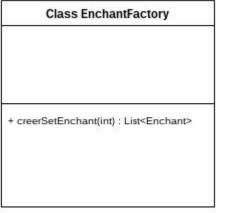


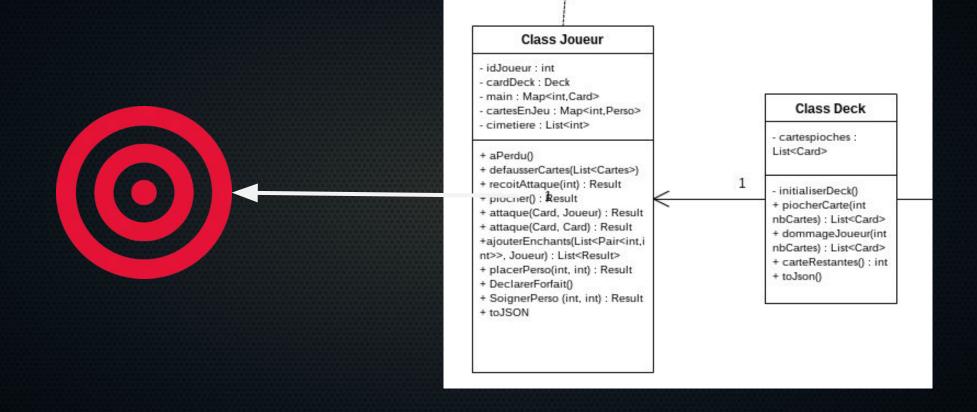
VII - Les factory





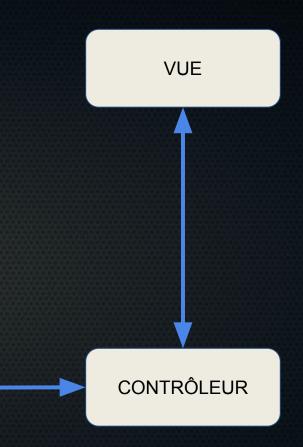




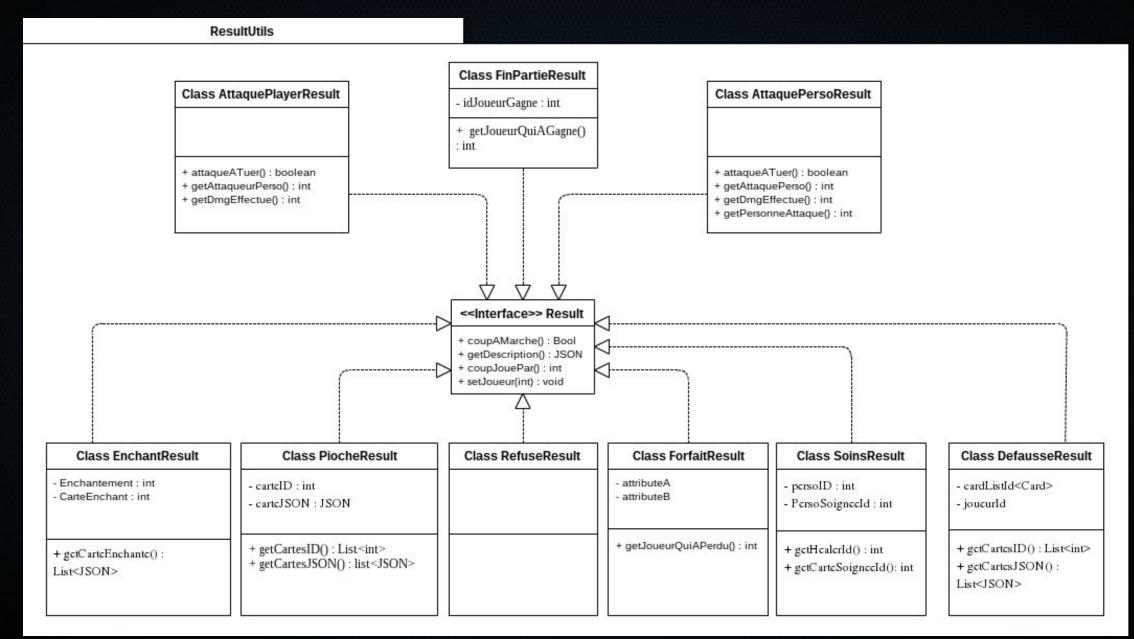


IX - L'API jeux

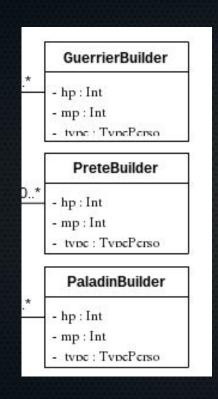
Class Jeux joueurList : List<Joueur> partieEnMarche: boolean joueurTour: int + prochainJoueur(): void + aQuiLeTour(): int + demarrerPartie(int): void + declarerForfait(int) : Result + getEtatJeu() : JSON + partieFini(): boolean + getJoueurGagnant(): int + piocherCartes(int) : Result + peutPiocherCartes(int) : boolean + attaquePerso(int, int, Carte, Carte) : Resultat + attaquePersoValide(int, int, Carte, Carte): Resultat + attaqueJoueur(int, int, Carte): Result + ajouterEnchantements(int, Carte, List<Carte>): Resultat + peutAjouterEnchantements(int, Carte, List<Carte>): + getCartesMainJoueur(int) : Collection<Carte> + getCartesJeuJoueur(int) : Collection<Carte> + getCimetiereJoueur(int) : Collection<Carte> + getJoueur(int) : Joueur + placerPerso(int, Carte, Carte, List<Carte>) : Resultat + peutDeployerPerso(int, Carte, Carte, List<Carte>): + defausserCarte(int, List<Carte>) : Resultat + peutDefausserCarte(int, List<Carte>): boolean + soignerPerso(int, Carte, Carte): Resultat + peutSoignerPerso(int, Carte, Carte): boolean + finPartie(): void



X - Les resultats



XI - Les morts aux combats (de carte)



<<Enumeration>> TypePerso

- + Guerrier
- + Pretre
- + Paladin

XII - Remerciements pour les idées reprises

- Philippe Pépos Petitclerc et Mehdi Ait Younes : Soingeur / Cible / Combatant
- Zerrouk Radhia et Belarbi Faten : Combatant
- Abdelkarim Belkhir et Dylan Lebatteux : Combatant
- Rubin Jehan et Haas Ellen : Fonction de validation de l'API
- Kaamelott : Sloubi