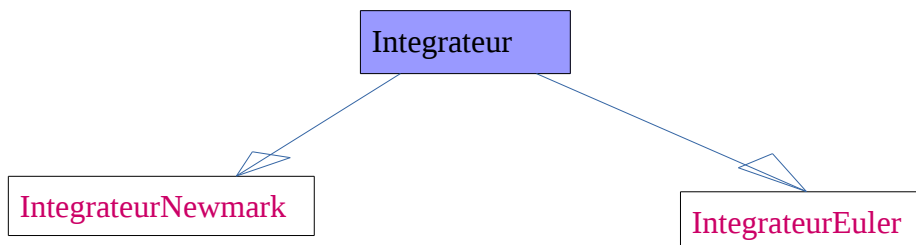
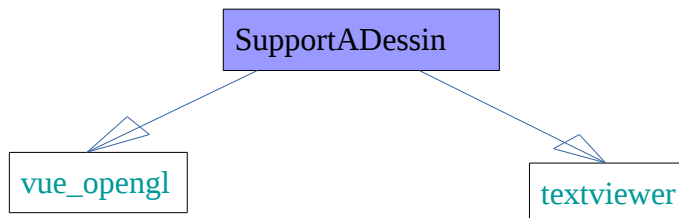
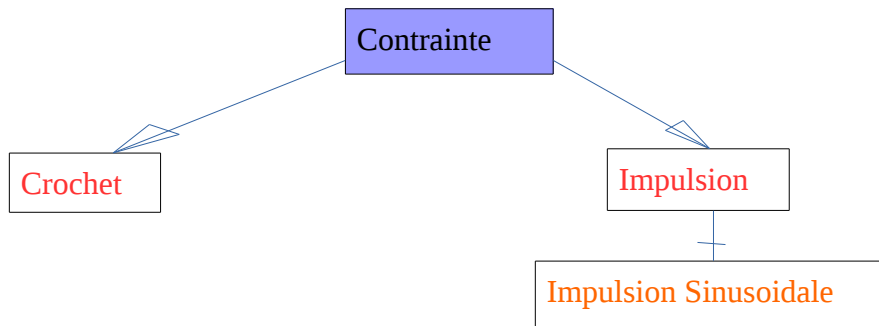
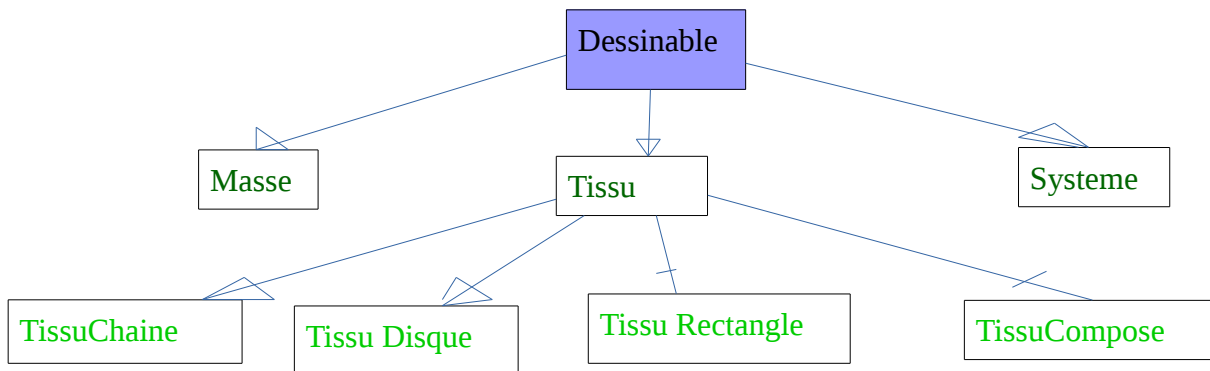




Conception



Légende :

-  Indique une relation d'héritage: elle pointe de la SuperClasse vers la classe héritée
-  Classe abstraite

Points de conception

Nous avons fait hériter une classe d'une superclasse à partir du moment où la sous-classe EST une superclasse.

De plus il est important de noter que :

- La Classe Dessinable a un pointeur sur SupportADessin
- TissuComposé a un tableau dynamique de pointeurs sur des Tissus géométriques
- Systeme a un tableau dynamique de pointeurs sur Contraintes et Tissus
- Impulsion a un tableau dynamique de pointeurs sur Tissus et Masses
- Masse a un tableau dynamique de pointeurs sur ressorts
- Nous avons décidé de compter le temps dans Systeme pour savoir si les contraintes sont bien arrivées à la fin de leur temps. Il y a donc un attribut non mentionné dans les consignes dans systeme : un double temps.
- Les Tissus ont des pointeurs sur Ressorts pour permettre de faire TissuCompose plus facilement.
- La conception de l'IntegrateurNewmark nous a forcé à créer une méthode supplémentaire dans Masse: `get_force_contraintes()` (cf code commenté).
- La methode `check()` dans la classe Tissu nous a également fait créer de multiples méthodes de Masse et Ressort, repertoriées après le commentaire `//Action sur les masses/ressorts`.
- Au sujet des destructeurs : nous n'avons pas fait de destructeurs dans Masse qui détruisent les Ressorts associés (avec leurs pointeurs) car nous considérons qu'il n'y aura que des TissusGeometriques et non des Tissus créés à la main. Les destructeurs de TissusGéométriques (mise à part celui de TissuCompose) détruisent tous leurs ressorts et toutes leurs masses avec les pointeurs associés. Le destructeur de TissuCompose détruit les ressorts qui ont été créés spécialement pour lui (ceux qui connectent les Tissus). Ses TissusGéométriques seront détruits par l'appel de leurs destructeurs dans le main.
- Pour faciliter le dessin des ressorts et axes, nous avons ajouté les méthodes suivantes dans la classe `vue_open_gl` :
 - `void dessine_ligne(Vecteur3D, Vecteur3D, double, double, double);` (depart, arrivee et R, G, B)
 - `void dessine_ressorts(Tissu const& t);` (dessine des lignes en parcourant les ressort du Tissu t)et dans la classe Ressort :
 - `double force() const;` , qui renvoie la norme de la force de rappel engendrée par le ressort en question