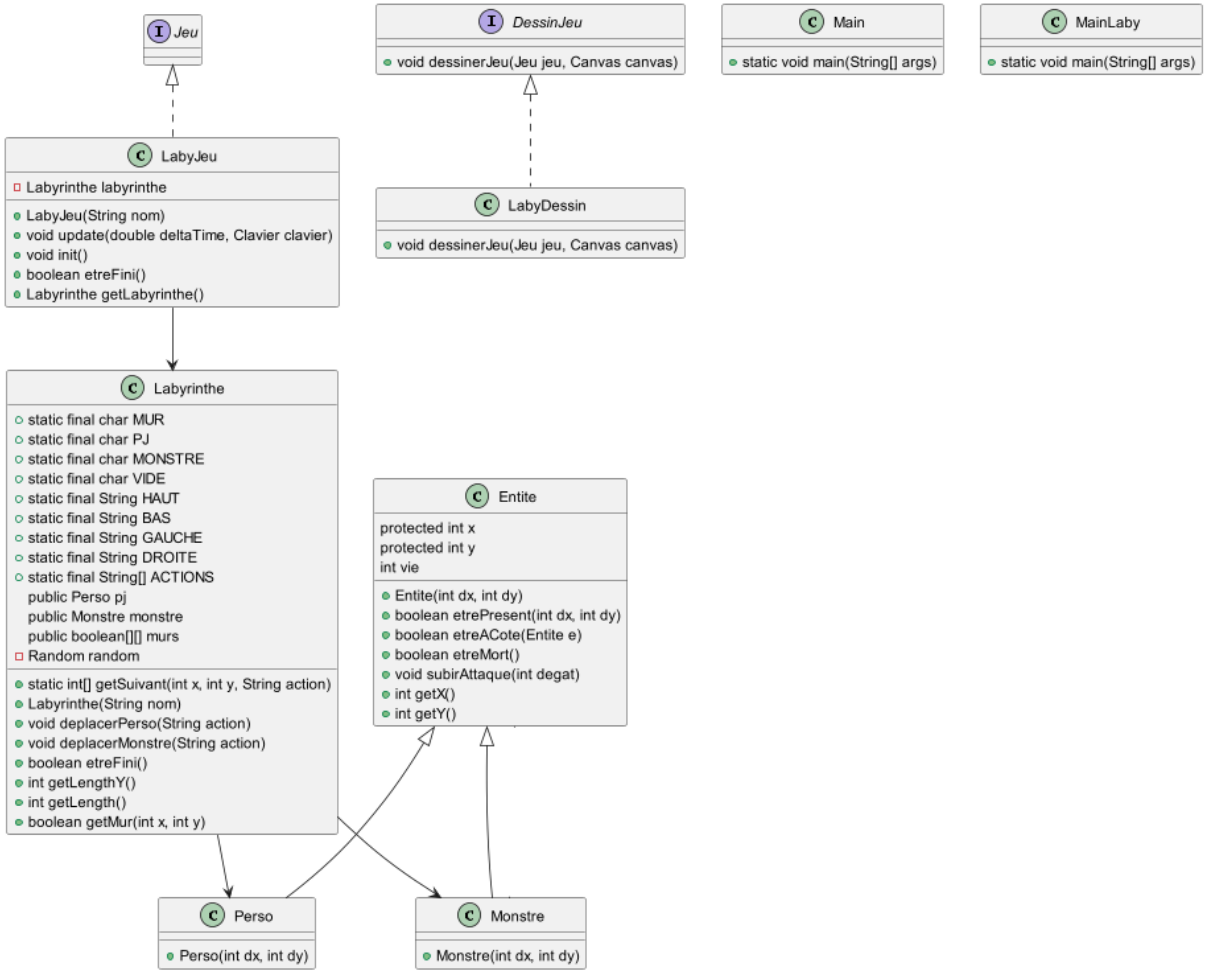
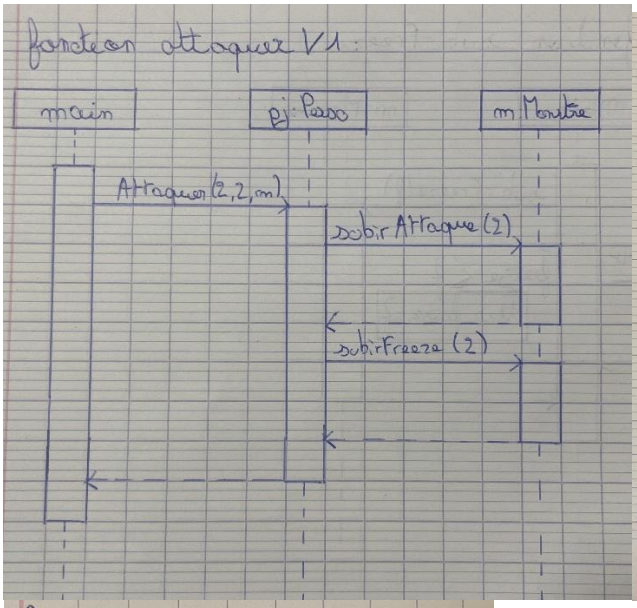
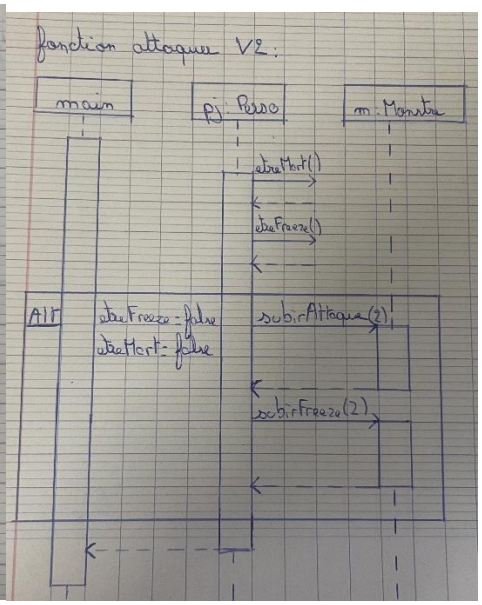
Dans les premières minutes avant notre seconde itération, nous avons d'abord eu un entretien avec notre professeur. Suite a ses conseils, nous avons consacrés quelques minutes à la modification de notre diagramme de classe. Il n'y eu aucun ajout, simplement le déplacement de la classe Perso comme enfant de Entité.
Nous avons donc refais ce diagramme :

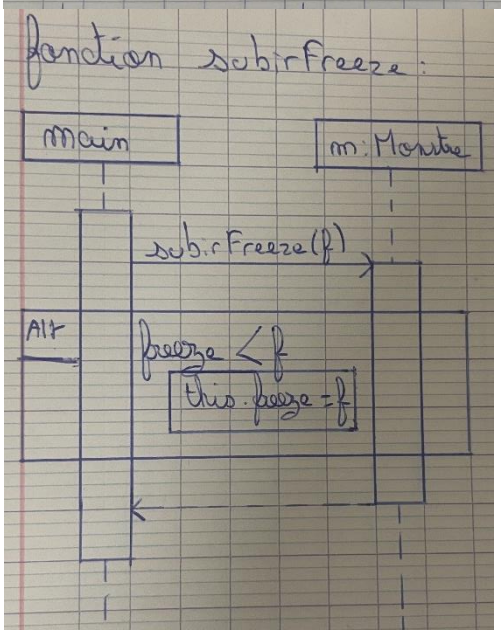


Au début de l'itération numéro 2, nous allons décider de prendre un chemin spécial avec notre jeu. Le but final ne sera pas de tuer le monstre mais de fuir vers une sortie qui sera faire dans les prochaines itérations. On essayera donc par la suite de faire plusieurs niveaux avec de plus en plus de monstre et une difficulté croissante.
Afin de fuir le/les monstres, nous avons rajouter un variable "freeze" qui va permettre au personnage d'attaquer le monstre et fuir pendant 1 étape.

Pour ce faire, nous avons d'abord réfléchi aux méthodes et fonctionnalités à rajouter dans les classes. Pour arriver à nos fins, nous avons élaborés des méthodes et modifiés d'autres pour que comme dit précédemment, le personnage puisse retarder le monstre et non pas le tuer.
L'élaboration de ces méthodes se sont fait de la même manière que pendant la première itération, nous avons d'abord écrit les diagrammes de séquence sur un tableau afin de se mettre d'accord avant de recopier au propre. En commençant par « subirFreeze » qui permettra de ralentir une entité et Attaquer qui permet donc d'attaquer.







Une fois les diagrammes dessinés, nous nous sommes attaqués au code, nous avons dû modifier trois classes : Labyrinthe, Entité et Clavier. Nous avons d'abord fait des modifications sur la classe Labyrinthe :

Nous avons commencé par créer un tableau d'entité.

Premièrement nous avons ajouté une méthode "deplacerEntite" pour ne pas copier-coller le code du déplacement (auparavant le code était copier-coller pour le monstre et le personnage). Cette méthode vérifie si notre entité est freeze avant quoi que ce soit. Puis elle vérifie si notre action n'est pas à nul (si l'action est nul, alors on met un déplacement aléatoire). Ensuite elle utilise le même fonctionnement pour prévoir le déplacement. Et pour tester si le déplacement est possible on va cette fois balayer le tableau des entités (cela nous permettra d'ajouter facilement des entités sans modifier nos méthodes).

Puis nous avons renommez la méthode "deplacerPersonnage" en "réaliserEtape", car dans cette méthode on gère les déplacements de toutes les entités, ainsi que toutes les attaques. On vérifie donc si l'action est un mouvement puis on bouge toutes les entités. Si c'est une attaque on déplace aucune entité et on effectue l'attaque du perso (si le perso n'attaque personne, il est freeze 1 étape). Et ensuite on balaye tous les monstres et si le joueur est dans son rayon on l'attaque.

Ces changements vont facilement nous permettre de rajouter des actions et des entités !

Pour conclure, après toutes ces modifications, nous avons fais le diagramme de classe final afin de répertorier tous ces ajouts.