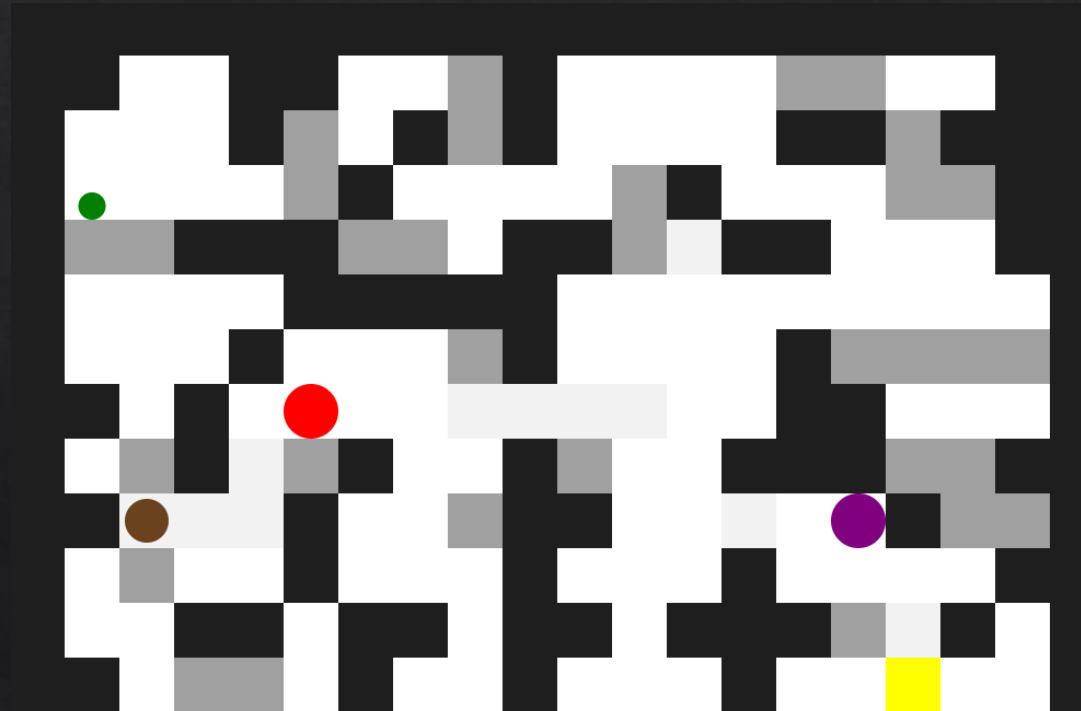


Jeandidier Clément
Hary Alexandre
Dirand Félicien
Jean-baptiste Evan

SAE 2.01 Développement d'une application

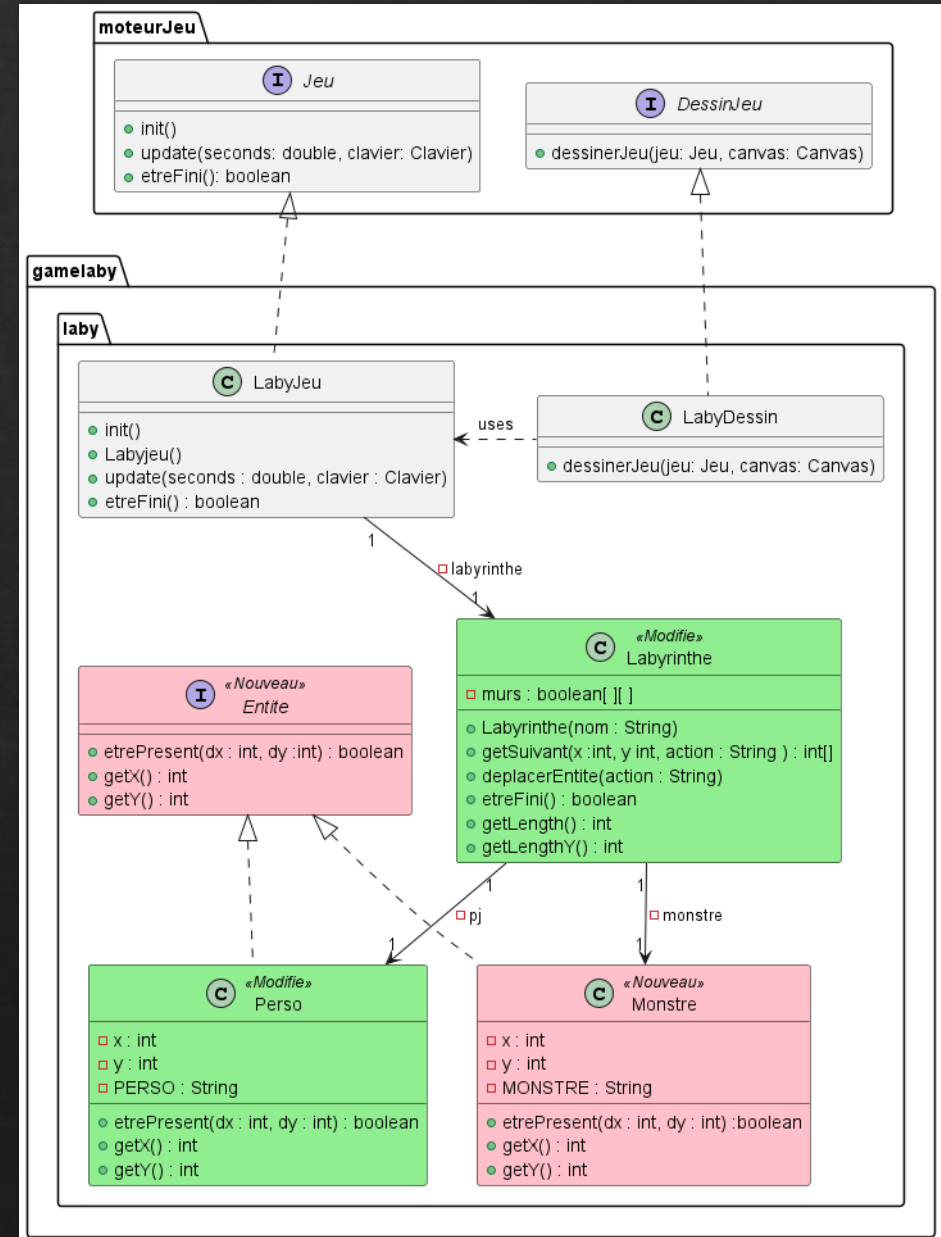
Jeu de type Bomberman



Présentation des différentes versions

(version 2)

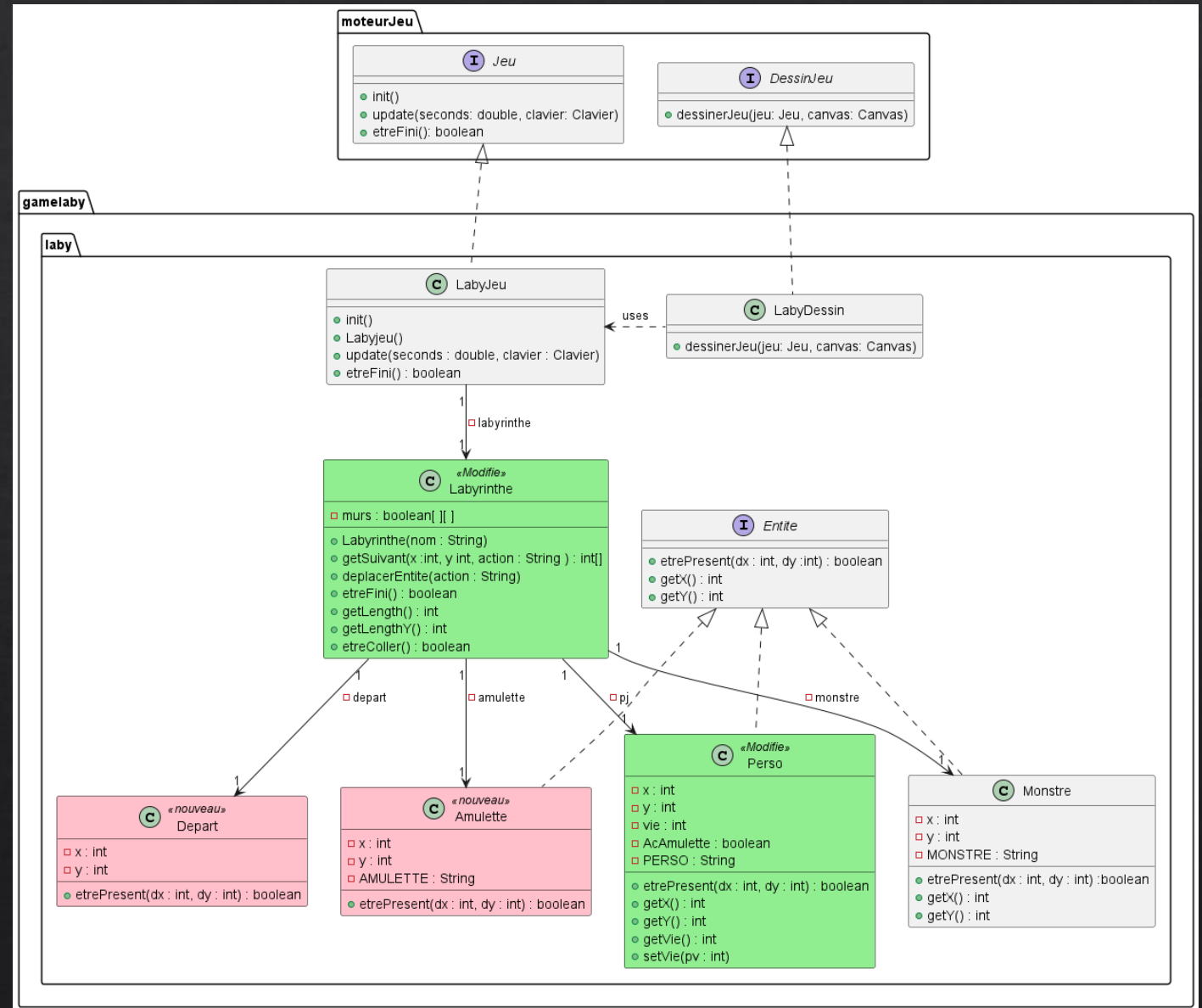
- Donner une position initiale au monstre
- Afficher le monstre
- Considérer le monstre dans les déplacements du personnage
- Déplacer le monstre (optionnel en fonction avancée)



Présentation des différentes versions

(version 3)

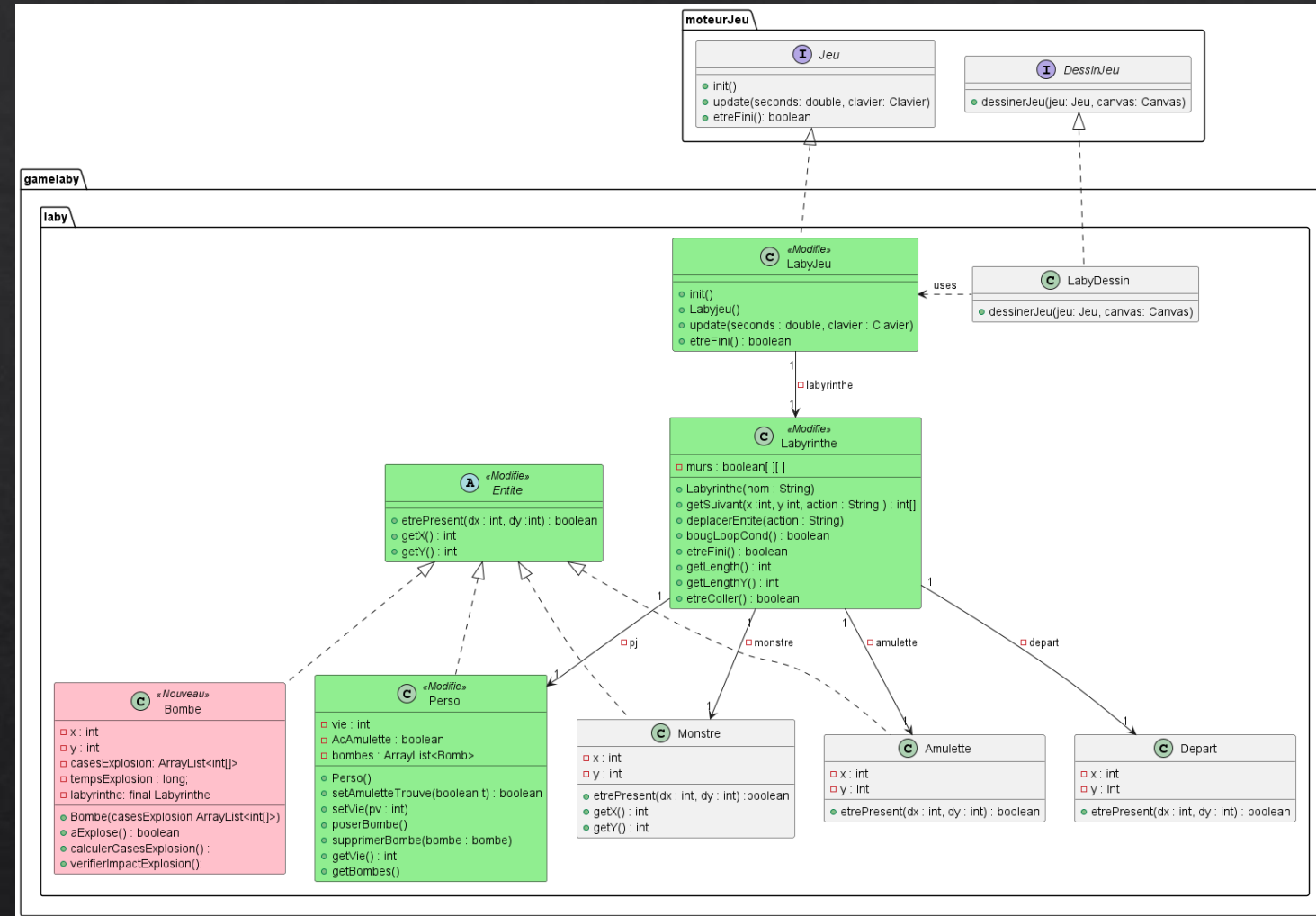
- Fin du jeu : mort du héros
- Fin du jeu : victoire du héros
- Attaque des monstres
- Mise en place de l'amulette
- Acquisition de l'amulette



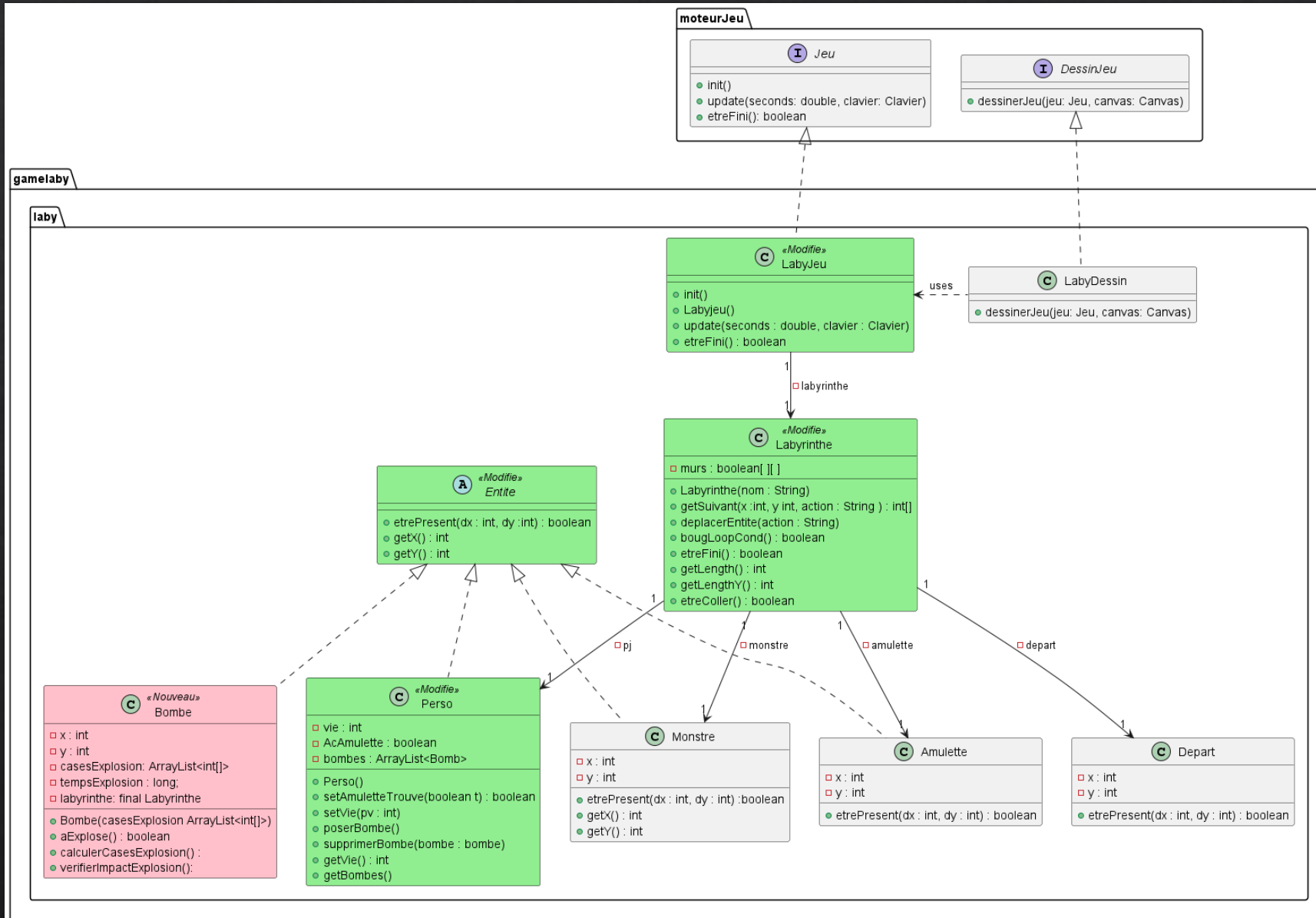
Présentation des différentes versions

(version 4)

- Dépôt d'une bombe
- Déclenchement de Bombes à blanc
- Dégâts d'une bombe
- Murs friables



Présentation de la conception finale



Présentation de la conception finale

Version 2 :

- Alexandre a initialisé la position initiale du monstre, il a créé la classe monstre, il a ajouté un attribut monstre dans la classe labyrinthe, dans le constructeur de cette classe, il a ajouté une condition qui fait que les coordonnées du M dans le labyrinthe soient les coordonnées du monstre.
- Evan s'est chargé d'afficher le monstre sur l'application sous forme d'un cercle violet
- Félicien a géré les collisions entre le personnage et le monstre, il a modifié `deplacerPerso()` en faisant une condition qui empêche le personnage d'être sur la même case que le monstre
- Clément s'est chargé du déplacement aléatoire du monstre qui se fait à chaque déplacement du joueur.
Il a du modifier `deplacerPerso()` et `deplacerEntite()` dans cette méthode , il a ajouté le déplacement du monstre.

Présentation de la conception finale

Version 3 :

- Alexandre a fait la méthode être fini qui arrête le jeu si le personnage n'a plus de vie, dans MoteurJeu, il a ajouté une condition qui arrête le jeu si êtreFini renvoie true
- Clement et Félicien ont fait le méthode recupererAmulette() et ont modifié la méthode êtreFini()

Dans le labyrinthe, ils ont ajouté une amulette que le personnage doit récupérer puis ramener à la case de départ

- Pour faire attaquer le monstre, Evan a rajouté une condition dans le déplacement du monstre qui fait que

le monstre va attaquer le joueur au lieu de se déplacer si le joueur est à côté de lui

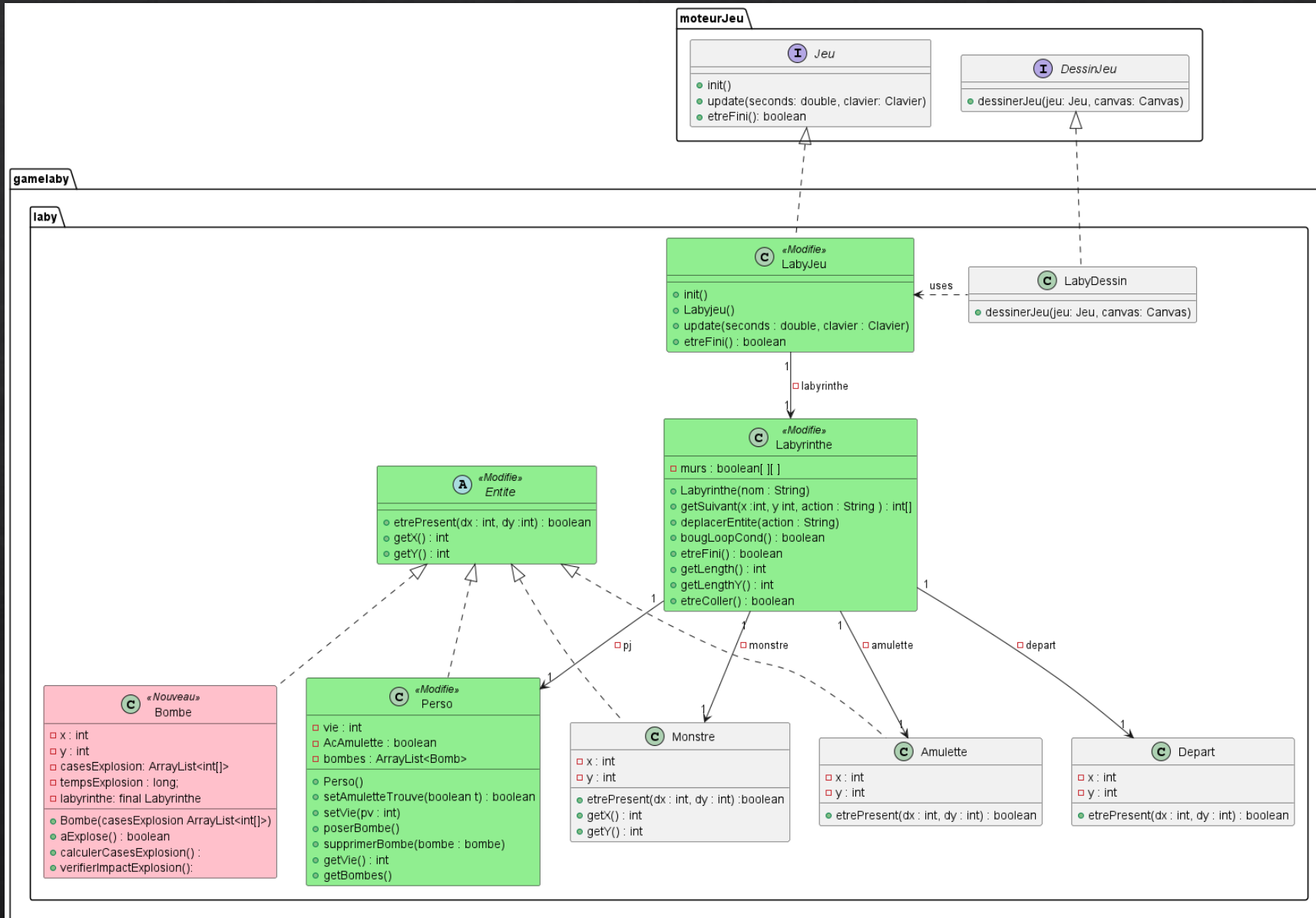
Dans Labyrinthe, Evan a rajouté une méthode être collé si renvoie true si le monstre est collé au personnage

Présentation de la conception finale

Version 4 :

- Clement et Felicien on cree les fonctionnalites de depotbombe , degatbombe, declenchement bombe
depot bombe a ete cree en appuyant sur la touche espace la bombe est place et elle ne peuvent pas être place au meme endroit et ont une limite de 3
déclenchement de bombe a blanc permet de faire quand la bombe est place au bout de 3 seconde la bombe disparaît
ensuite dégât bombe complémente bombe a blanc en rajoutant des langue de feu qui permet de détruire les murs et de faire des dégâts au joueur
- Evan et Alexandre ont crée la fonctionnalité des murs friables
les murs friables ont été crée en héritant de la classe abstraite Entite et ont un attribut de destruction
Ils ont du de même mettre en place la prise en compte des murs friables dans le déplacement du perdu et du monstre

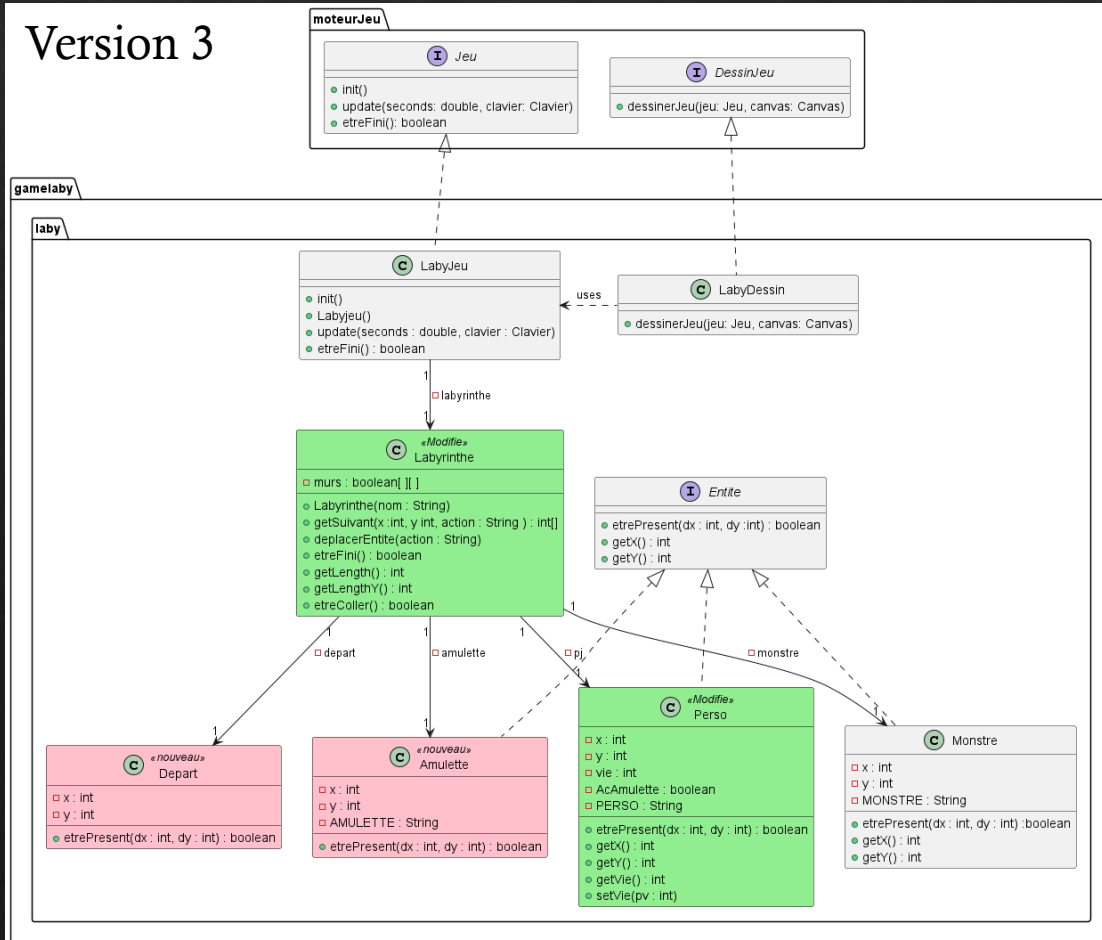
Présentation de la conception finale



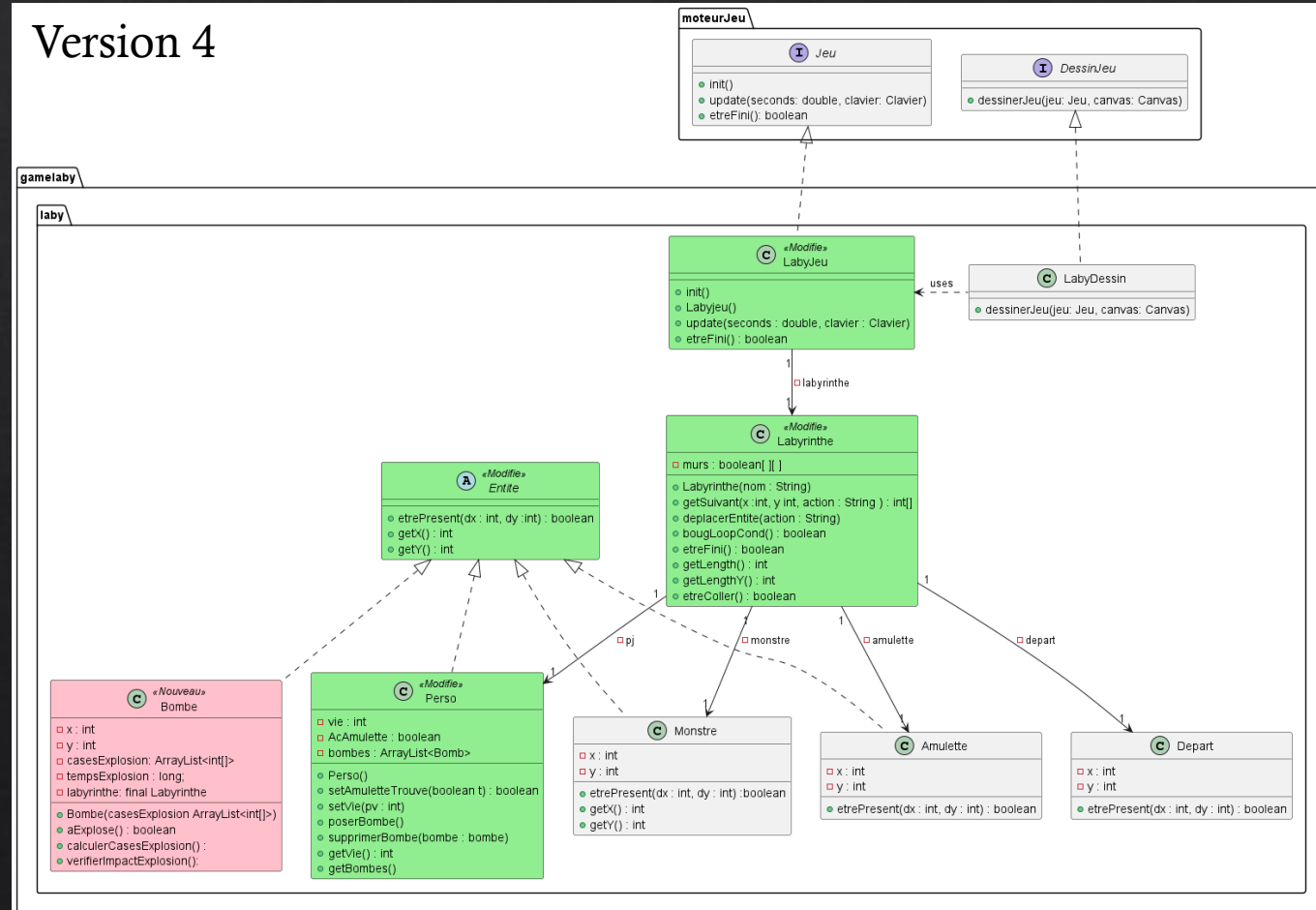
Présentation d'un point important

On a du passer Entite en classe abstraite

Version 3



Version 4



Présentation d'un point important

Une fonctionnalité qui nous a posé problème.

- Destruction d'un mur friable

