Dossier du Projet (19-20)

Groupe : Matéo Laville, Alexandre Khefif-Derain, Thomas Huxley-Neto

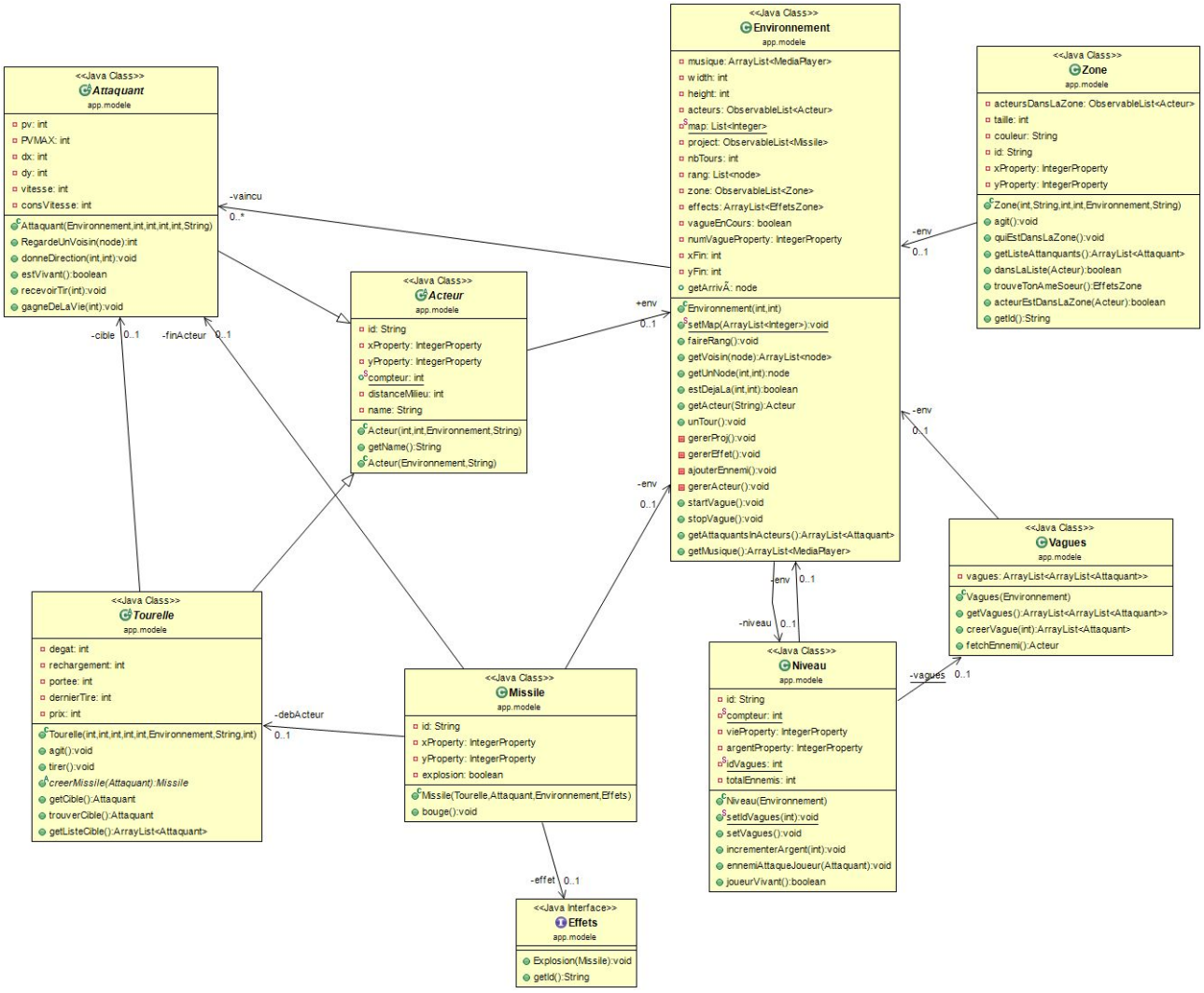
CPOO		IHM		Projet Tuteuré		Gestion de Projet	
diagrammes d'architecture	9	ergonomie et contenus	15	soutenance	10	document utilisateur	8
diagrammes de classes	9	programmation évènementielle et JavaFX	35	niveau et qualité du produit fini	15	Trello	6
diagrammes de séquence	9			notes de sprints	15	Git	6
POO	24						
Structures de données	6						
gestion des erreurs	5,5						
tests	5,5						
algos	17						
ampleur et qualités du code	25						
	110		50		40		20

Documents pour CPOO (sur 27)

1.1 Architecture (9 points)

Pour l'architecture notre équipe sommes parti sur un template assez simple. Nous avons une classe Environnement qui connait absolument tout de la partie (Niveau / Map /

Acteurs / etc) et presque toute les classes ont un Environnement dans leurs constructeur.

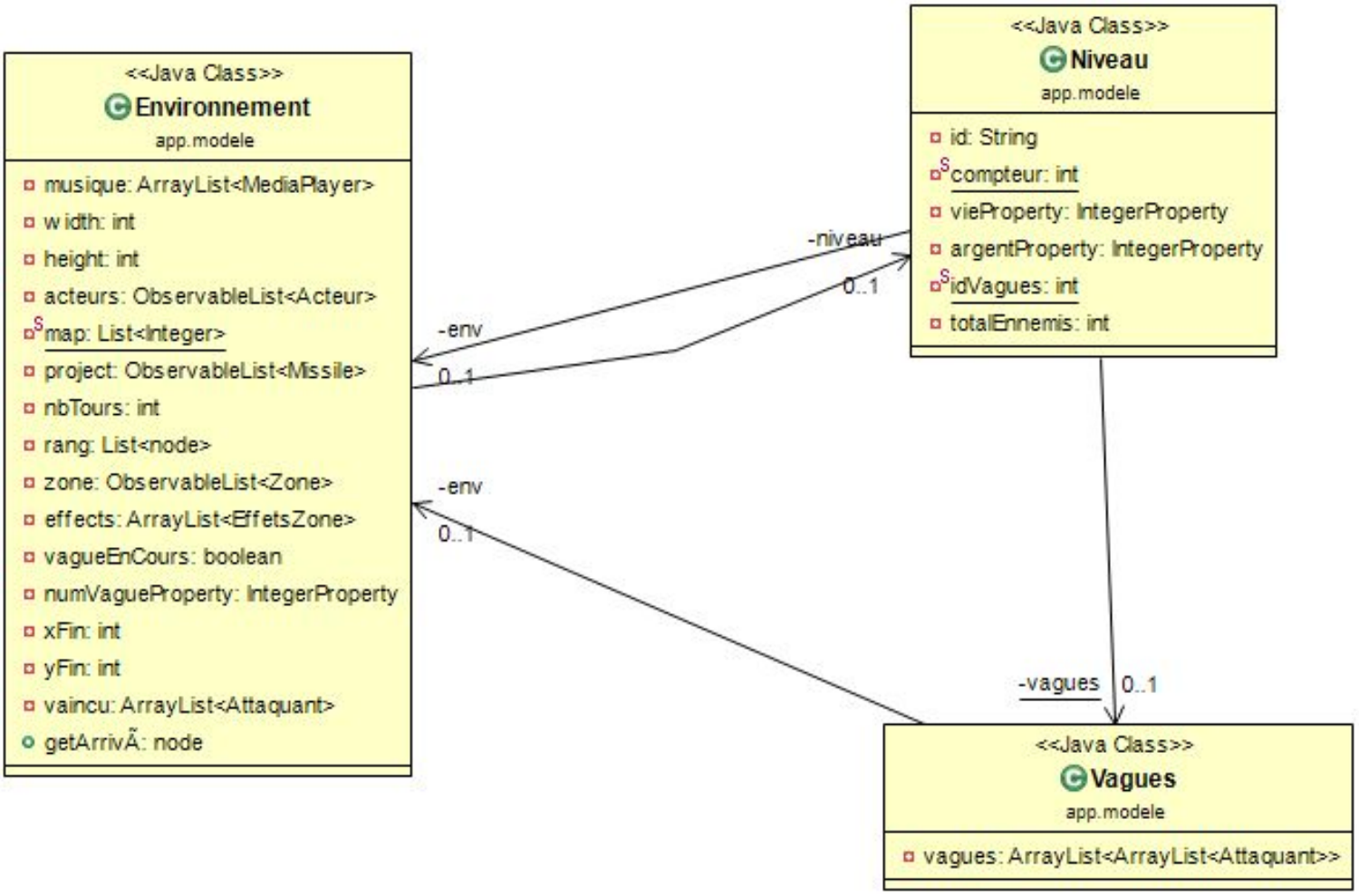


Donc dans ce diagramme de classe on retrouve bien notre environnement qui est la grande classe du projet. J'estime qu'à ce moment notre code se sépare en 4 parties : Les acteurs, les missiles, les niveaux et enfin les zones.

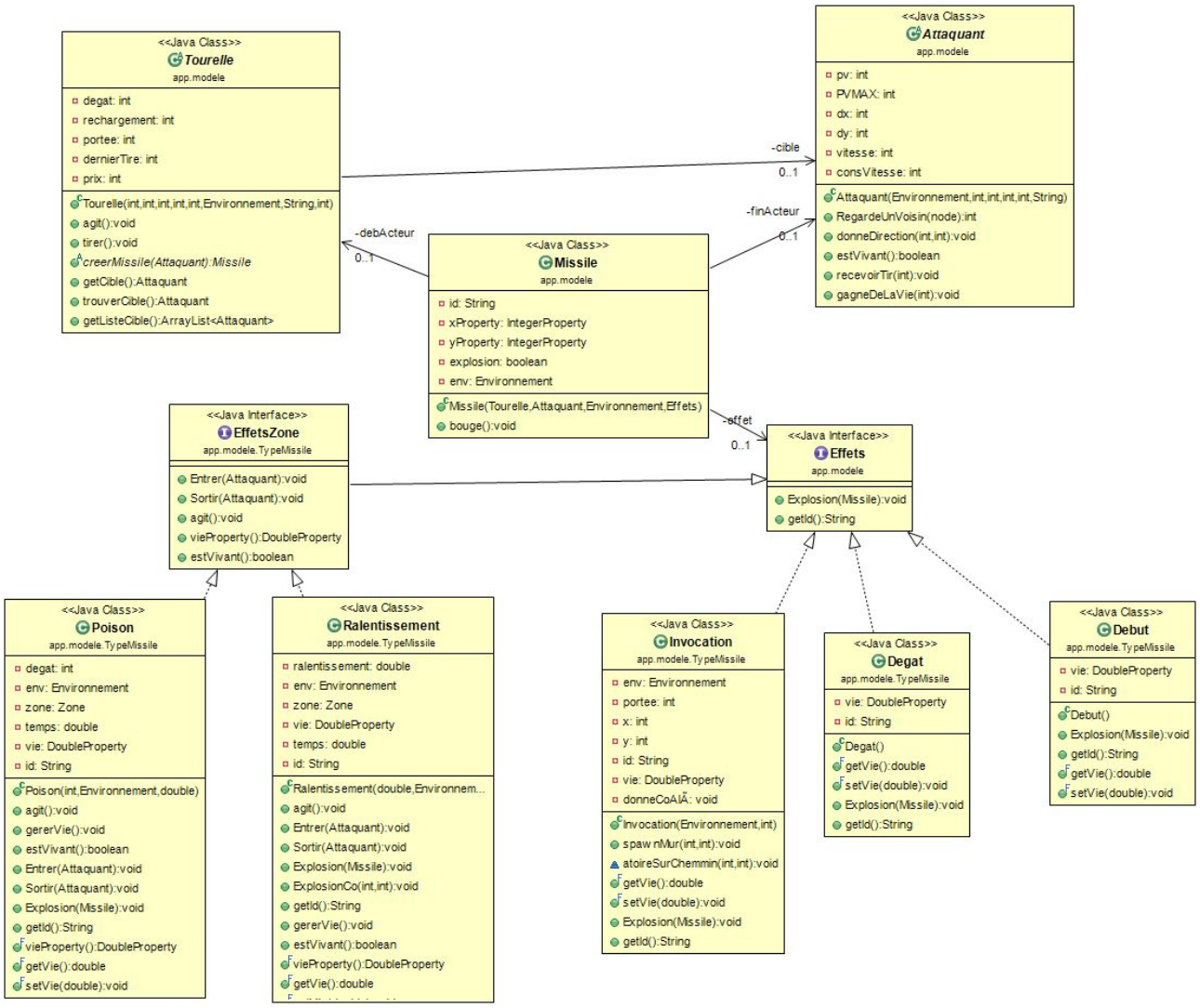
- La classe Acteur est la grande classe de tous les acteurs de la map (Attaquant, Tourelle ou encore Mur) donc toutes ces classes extends Acteur.

- Pour les missiles lorsqu'une tourelle lance un missile celui-ci contient un sort dès sa création. Lorsque le missile atteint sa cible il lance la méthode Explosion présente chez tous les effets (Après cela le missile disparaît de la map).A la suite de cela il y a deux possibilités si l'effet est un effet simple il disparaît juste après avoir effectuer explosion, mais si il s'agit d'un effet de zone il créer une zone qui vas détecter les entrée et sortie de la zone dans le cas ou elle détecte quelqu'un elle vas lancer une de ces deux méthode présente chez l'effet zone, Entrer() ou Sortir().

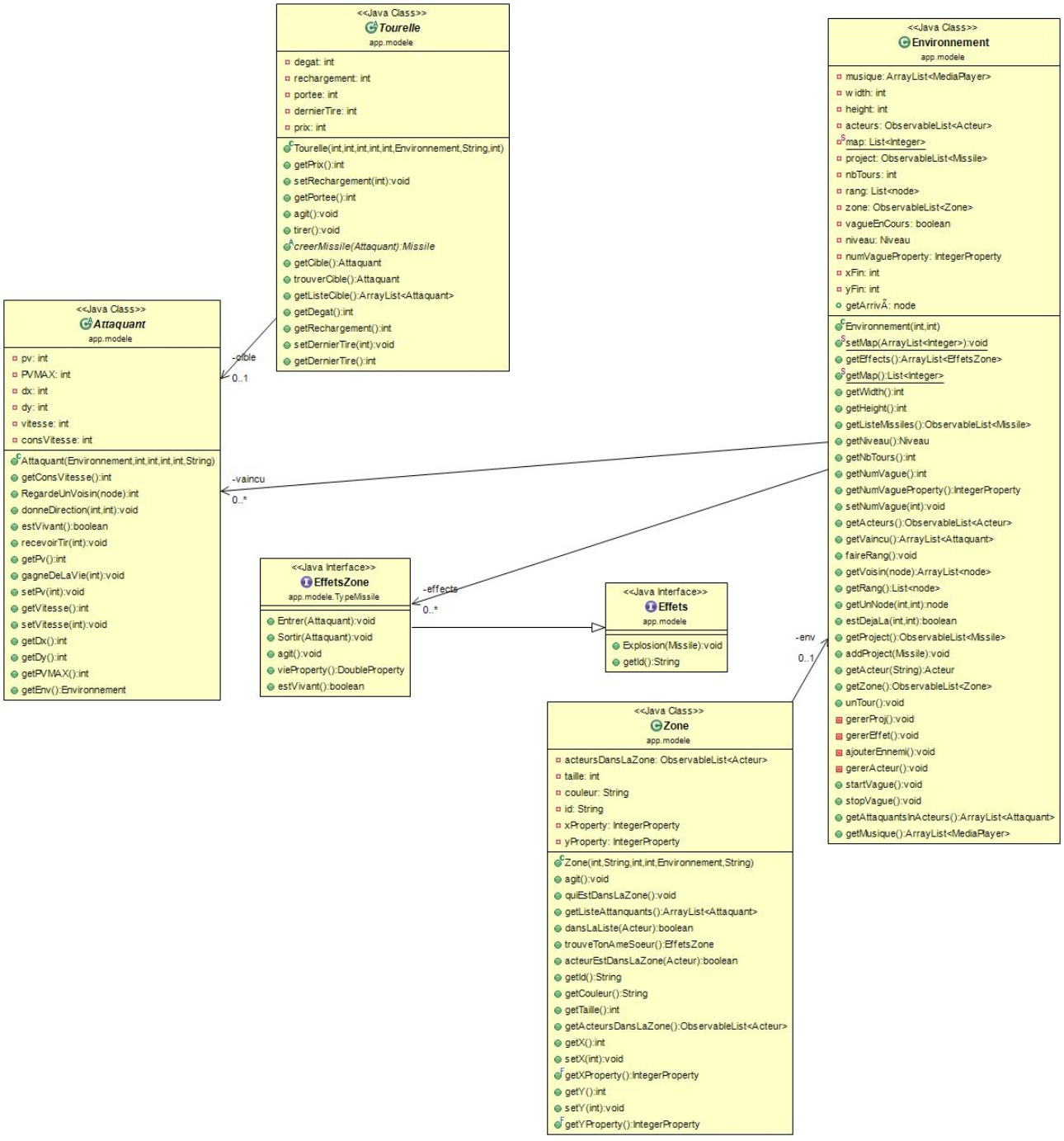
- Une vague a une liste à deux dimensions d'ennemis et un environnement, à chaque tour la vague envoie un ennemi à son environnement. Les vagues sont créées par un niveau qui contient la vie et l'argent du joueur ainsi qu'un environnement. Au final, Niveau sert à gérer les informations en liens avec le joueur et ce qui différenci les niveaux, Vagues et un outil pour créer et stocker des vagues d'ennemis, et Environnement utilise toutes ces informations pour fonctionner.



1.2 Détails : diagrammes de classe (9 points)



Je vais expliquer ici le fonctionnement des missiles / sortent et zone. Tour d'abord chaque tourelle a un temps de rechargement, dès que celle-ci a recharger elle fait spawn un missile. Le missile a besoin de l'acteur de départ, l'acteur de fin, un environnement et un effet. Ce missile est ajouté à la liste, et à chaque tour la méthode bouge de tous les missiles s'active et va calculer la direction à prendre pour atteindre l'acteur de fin. Dès que les coordonnées du missile sont égales à celles de l'acteur de fin il disparaît et active la méthode explosion de l'objet effet. Il existe deux types d'effets : les effets normaux et les effets de zone. Cette zone va avoir pour objectif de détecter les nouveaux membres de la zone, ceux qui rentre et ceux qui sortent. Tous les acteurs présents dans la zone sont dans une liste de la classe Zone.



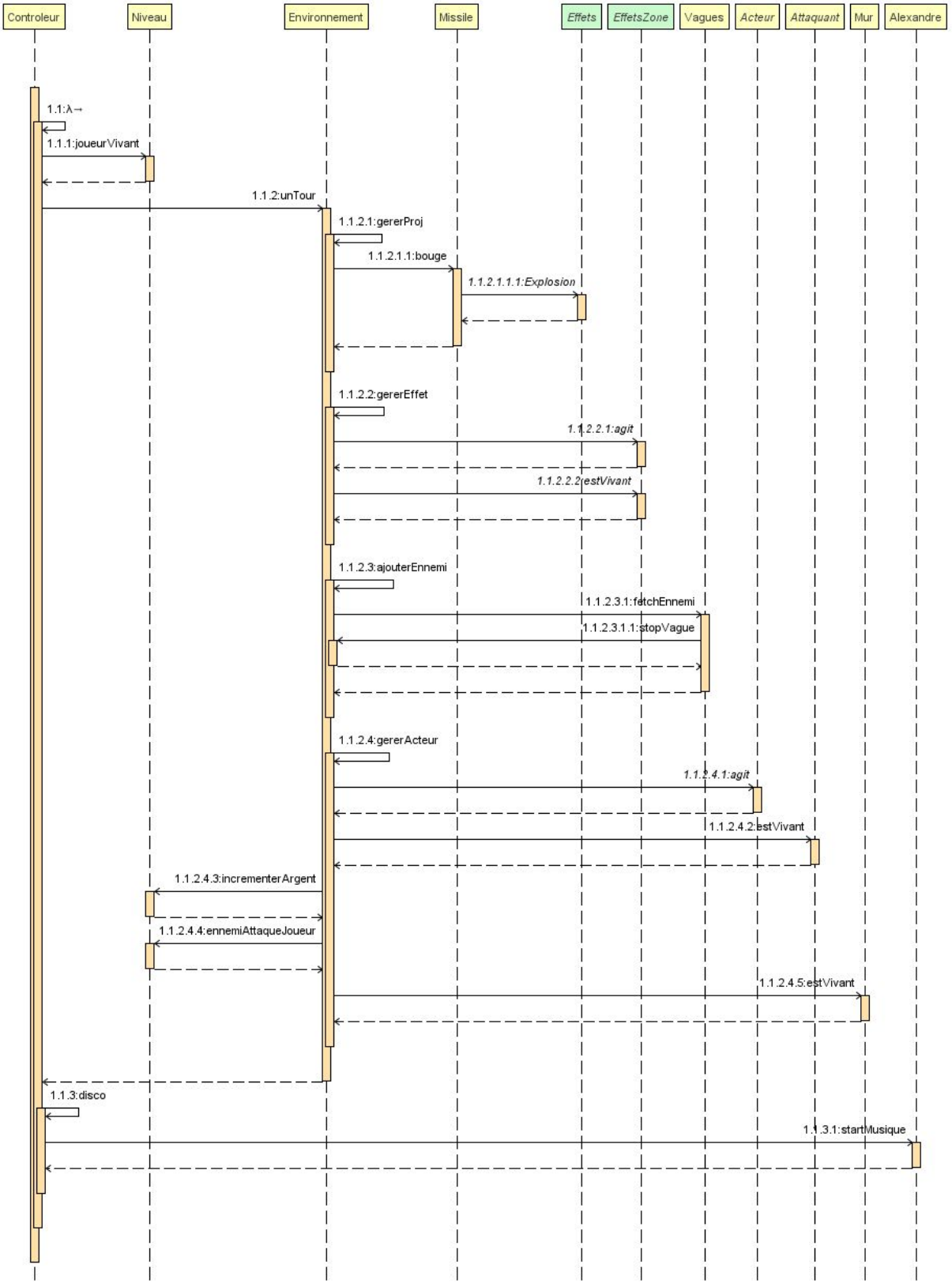
1.3 Diagrammes de séquence (9 points)

La méthode initAnimation() du contrôleur sert de game loop, effectuée toutes les 0.07 secondes.

On commence par vérifier si la partie est terminée : si le joueur est mort ou s'il n'y a plus d'ennemis en réserve.

Si la partie n'est pas fini, on appelle la méthode unTour() de l'Environnement qui va s'occuper de gérer les différentes entités s'y trouvant : en commençant par les projectiles (gererProj()) qui va les faire bouger un par un et vérifier s'ils ont atteint leur cible (si c'est le cas il explose, Explosion() dans Effets);

on s’occupe ensuite des effets, chacun agit, et disparaît s’il n’est plus vivant;
si une vague est en cours, ajouterEnnemi() appelle fetchEnnemi() dans Vagues, puis stop la vague si elle est vide (stopVague() dans Environnement);
vient enfin le tour des acteurs, les attaquants agissent et disparaissent s’ils sont morts, l’argent est alors incrémenté, et s’il sont arrivés à la fin ils attaquent le joueur, on retire ensuite les murs morts.
De retour dans le contrôleur, on appelle la méthode disco() qui, s’il y a un attaquant Alexandre ou un nyan cat (Telio nommé “Nyan”) sur le terrain, appel la méthode startMusique qui lance la musique de Nyan Cat (si elle n’est pas déjà lancée), et fait clignoter l'écran de toute les couleurs.


Controleur
Niveau
Environnement
Missile
Effets
EffetsZone
Vagues
Acteur
Attaquant
Mur
Alexandre
1.1:λ→
1.1.1:joueurVivant
1.1.2:unTour
1.1.2.1:gererProj
1.1.2.1.1:bouge
1.1.2.1.1.1:Explosion
1.1.2.2:gererEffet
1.1.2.2.1:agit
1.1.2.2.2:estVivant
1.1.2.3:ajouterEnnemi
1.1.2.3.1:fetchEnnemi
1.1.2.3.1.1:stopVague
1.1.2.4:gererActeur
1.1.2.4.1:agit
1.1.2.4.2:estVivant
1.1.2.4.3:incrementerArgent
1.1.2.4.4:ennemiAttaqueJoueur
1.1.2.4.5:estVivant
1.1.3:disco
1.1.3.1:startMusique

1.4 Structures de données :

La map est une liste static d'Integer, chaque texture corresponde à un numéro afin d'être affichées sur un tile pane, les acteurs (tourelle, attaquant, mur, missile) sont stockés dans une liste d'acteurs dans Environnement et s'affichent sur un pane, on se sert d'hashmaps (skins, skinsMissiles) dans le contrôleur afin de pouvoir leur attribuer un skin en fonction de leur nom (attribut name). Les statistiques du niveau sont calculées à la fin (à l'exception de la liste totale des ennemis vaincus) de la partie pour les envoyer à l'écran des scores. Chaque ennemi se trouve dans une vague du niveau en attendant d'être envoyé dans l'environnement.

1.5 Exception :

Contrôleur : La sélection de la tourelle se fait en calculant la position du curseur par rapport à la vbox, en divisant sa position par 80 on arrive ainsi à distinguer chaque image pour les sélectionner, mais ce calcul n'étant pas une science exacte il arrive que le curseur soit interprété comme étant sur une 9ème image (inexistante), au moment d'aller chercher cette image dans la liste il y a donc une erreur Out of bound exception, le try catch la récupère et indique dans la console quelle image le programme a interprétée.

BFS : Lorsque les ennemis se déplacent, il peut arriver que certains d'entre eux sortent du terrain et créait ainsi une erreur, à ce moment-là le try catch appel une méthode getMilieuChemin qui lui renvoie les coordonnés (sous forme d'un tableau) du milieu du chemin le plus proche. Néanmoins cette erreur venant certainement d'une erreur de programmation, nous avons réglé le problème en amont en obligeant les attaquants à toujours rester à une certaine distance du milieu du chemin.

1.6 Utilisation maîtrisée d'algorithmes intéressants :

Dans cette partie nous allons parler de notre algorithme BFS. Celui-ci fonctionne en 2 étapes. La première permet d'initialiser les nodes du terrain. Une node c'est une tuile du terrain chaque node a en paramètre ses coordonnées X et Y mais aussi la distance par rapport à la node de fin. Donc à la méthode faire rang notre programme regarde les tuiles adjacentes à celle de fin (qui a été trouvé grâce à son numéro d'identification) , si la tuile est disponible donc pour marcher (Reconnaissable grâce à son numéro de tuile paire si l'attaquant peut marcher dessus) et que la tuile n'a pas déjà une node associer dans la liste (avec les mêmes coordonnées) alors nous allons fabriquer un nouvel objet de type node que nous allons stocker dans la liste des nodes. Toutes ces manipulations se trouvent dans une boucle fort donc elle fabrique tout le chemin sous forme de node.Enfin nous allons voir comment les attaquants utilisent le BFS. Tout d'abord ils calculent les coordonnées de leur node actuel puis il regarde les nodes adjacentes et en fonction de celle avec la distance la plus faible ils calculent la direction à prendre pour se rapprocher de cette node

1.7 Junits :

Zone: Dans ces tests Junit on vérifie que la détection des entrées sorties marche. Pour cela nous avons 3 test (Avec un before):

- Le premier permet de vérifier si un attaquant spawn directement dans la zone si il est bien détecté.
- Le deuxième si un attaquant quitte la zone
- Le troisième permet de vérifier lorsqu'un attaquant se fait tuer dans le zone si sa mort et donc sa disparition est bien détecté.w²

2 Documents pour Gestion de projet

2.1 Document utilisateur (8 points) :

- Description du jeu (son objectif, son univers…) :

Tout d'abord, l'univers de notre Tower Defense se base sur l'univers de notre IUT, on pourra reconnaître certaines personnes telles que des professeurs de l'IUT qui auront le rôle des tourelles dans un Tower Defense et on pourra aussi reconnaître des élèves de l'IUT qui auront le rôle des ennemis qui avancent sur le terrain pour arriver jusqu'en bout de map.

Ensuite, l'objectif de notre jeu où l'on incarne le rôle des professeurs est de faire en sorte que les élèves n'atteignent pas la fin de l'année (représenté ici comme dans la plupart des Tower Defense avec un chemin que l'ennemi parcours et une destination atteindre.)

-Description des tours et des ennemis (force et faiblesse les une par rapport aux autres , tableau de relation).

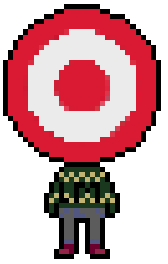
Liste des professeurs:

-Tour n°1 M Bonnot (Philippe Bonnot étant un des premiers professeurs que nous avons rencontrés lors de notre année scolaire nous avons voulu le représenter comme étant aussi une des premières tours disponibles en jeu. Ce professeur envoie sur les élèves des livres d'algèbre. Grâce à sa flexibilité de directeur d'étude il a une portée d'attaque moyenne mais aussi une attaque dans la moyenne. Le coût de Monsieur Bonnot est plutôt faible pour que cette tour soit accessible dès le début du jeu).

-Tour n°2 M Bossard (Aurelien Bossard étant notre professeur de bas niveau mais aussi étant connu par ses tirages aux sorts nous avons voulu lui attribuer comme caractéristique la possibilité d'attaquer aléatoirement un ennemi dans une zone voulut, son temps de rechargement est plutôt moyen. Grâce à sa grande taille il a une portée bien plus grande que M Bonnot. Monsieur Bossard envoie des craies sur les élèves tout en ayant un coût un peu plus élevé que Monsieur Bonnot).

-Tour n°3 Mme Clément-Comparot (Véronique Clément-Comparot est une professeur multifonction car celle-ci a fait plusieurs matières différentes durant notre premier et second semestre. Elle est reconnue internationalement grâce à ses memes légendaires elle aura donc la capacité de faire rire les élèves causant une déconcentration massive des élèves impliquant une diminution non négligeable de la vie des élèves. Son coût est plutôt moyen par rapport à Monsieur Bonnot).

-Tour n°4 Mme Lamolle (Myriam Lamolle est notre professeur depuis nos début mais aussi depuis peu notre nouvelle directrice de l'IUT de Montreuil. Nous avons donc voulu l'intégrer dans le jeu, afin de garder un lien logique entre le jeu et les évènements de l'IUT nous avons voulu lui attribuer le rôle de "Sniper", elle a donc la capacité de faire un nombre de dégâts non négligeables avec une portée illimitée mais son temps de rechargement est énorme. Elle a pour apparence une tête de cible car elle n'a pas pu nous fournir une image d'elle. Ses projectiles seront représentés par des commandes SQL (représenté par "SQL" écrit en rouge). Elle a un coût plus haut que la moyenne.).



-Tour n°5 Mme Ricordeau (Madame Ricordeau est notre professeur de mathématiques (Fonction) Elle est assez connu pour ses digressions lors des amphithéâtres de mathématique entraînant parfois des retards sur le cours tout en faisant sourire les étudiants bien évidemment. Elle a la capacité de ralentir les élèves et de les étourdir sur une zone tout entière. Ses munitions sont des textes du type "BlaBlaBlaBlaBlaBla" elle ne peut faire aucun dégât sur les élèves mais peut les ralentir comme dit précédemment dans une zone. Son prix est dans la moyenne.).

-Tour n°6 M Homps (Marc Homps mathématicien de renom étant presque détenteur de la médaille Fields. Ce professeur pourra faire de faibles dégâts aux étudiants mais il a une capacité exceptionnelle. Il a un certain pourcentage de chance de clasher un élève entraînant un retour à la case départ pour celui-ci. Sa portée est grande mais il a un coût non négligeable.).

-Tour n°7 M Rety (Jean-Hugues Rety professeur JAVA que nous avons côtoyé plusieurs matins avant nos cours d'amphithéâtre, nous avons remarqué qu'il avait quelques habitudes / tic comme prendre son café tous les matins ou encore enlever et remettre ses lunettes approximativement 3 fois par seconde. Nous avons choisi de lui attribuer un rôle bien spécial en rapport avec un de ses tics. La tour Rety a une attaque modérée mais sa cadence de tir changera de manière régulière entre une cadence normale et une cadence très rapide. Sa portée est moyenne mais son coût est élevé.).

-Tour n°8 M Simonot (Marianne Simonot étant responsable du module Interface Homme-Machines elle a eu à affronter une vague immense d'étudiants ne comprenant pas un mot de java FX elle a donc eu besoin de mettre en place des barrières afin que la vague d'étudiants ne s'abatte pas sur elle d'un coup. C'est depuis ces jours sombres que la Tour simonot fut créée. Elle a donc la capacité de faire apparaître 3 barrières qui peuvent arrêter chacune un étudiant. Son coût est très élevé avec une portée dans la moyenne.).

<u>Liste des étudiants :</u>

- Étudiant n°1 Télio (Cet étudiant est tout simplement normal, il travaille normalement, il marche normalement, il vit normalement, bref il est normal. Il est donc l'étudiant qui sert de base par rapport aux autres car tous les étudiants lui ressemblent au minimum. Il aura une vitesse et des points de vie normaux.).

- Étudiant n°2 Haris (Haris est un des élèves les plus indéfinissables de cette promotion. Sa petite taille lui permet de se faufiler discrètement entre les personnes de l'IUT, nous lui avons donc attribué le rôle de "Ninja". Il n'est pas spécialement rapide mais quand il est la cible d'un tir, il a un certain pourcentage de chance d'éviter celui-ci et de continuer sa route. Par contre, il ne pourra pas éviter les tours Ricordeau et Comparot car il s'agit d'attaques de zone.).

- Étudiant n°3 Théo (Cet étudiant, avec son avance non négligeable sur le programme, a la capacité d'aider les étudiants autour de lui : les étudiants présents à une distance raisonnable de lui sont soignés. Malgré cela il aura très peu de vie et une vitesse moyenne).

- Étudiant n°4 Matéo (Cet étudiant fait partie du trio de boss TAM. Il pourrait peut-être nous rappeler une tortue due à son calme imperturbable et sa concentration hors norme. Ramenant tout le temps son produit hydro-alcoolique lorsqu'il sort de chez lui, il a appris à prendre très grand soin de sa santé et possède désormais une santé d'une qualité inégalable. Il est certes lent dû à son calme,

mais possède une quantité de points de vie monstrueuse, il ne vaut donc mieux pas qu'il atteigne la fin d'année. Mais si jamais le joueur parvient à éliminer Matéo, ce dernier invoquera plein de camarades pour qu'ils puissent atteindre la fin d'année à sa place.).

- Étudiant n°5 Thomas (Cet étudiant faut lui aussi partie du trio de boss TAM. Il a souvent été dans un rôle de support pour ses camarades et a donc pour caractéristique d'encourager les étudiants en leur donnant un bonus de vitesse. Il n'a pas autant de points de vie que Matéo, mais reste assez résistant et possède une vitesse relativement élevée).

- Étudiant n°6 Alexandre (Cet étudiant est le dernier élément du trio de boss TAM. Grâce à son implication monstrueuse dans tout ce qu'il fait, il a transcendé tous les TP et a acquis des compétences encore inconnues actuellement. Il a réussi à invoquer ce qui semble être des chats avec une vague d'arc-en-ciel derrière eux et ayant un corps rectangulaire rose. Pourquoi des chats ? Parce qu'il aime ça. Pourquoi une vague d'arc-en-ciel ? Parce qu'il aime ça. Il réussit même à hacker les enceintes de l'IUT pour faire tourner une musique plutôt spéciale. Pourquoi cette musique ? Parce qu'elle lui permet une très bonne concentration, sans cette musique, il n'aurait jamais pu atteindre le niveau qu'il a obtenu aujourd'hui. Bref c'est Alexandre, c'est un boss, donc il possède plus de vie que la moyenne, il possède une vitesse normale et peut invoquer des chats qui l'aideront à atteindre la fin de son année scolaire.).

Scoring :

Concernant le scoring, on a décidé de viser très simple. À la fin de la partie, que le joueur gagne ou perde, il y aura tout simplement un tableau de score qui va apparaître sur l'écran renseignant tout d'abord s'il a gagné, le nombre d'ennemis qu'il a tués, la vie du joueur à la fin de la partie et le score (pourcentage de vie restant).