

# Documents Pour les SAEs S2 2022-2023

## 1. Auteurs

Estéban MARIE-LOUISE, Sofia LOPES, Oumaima ELKHADRAOUI, Joseph MIAKA

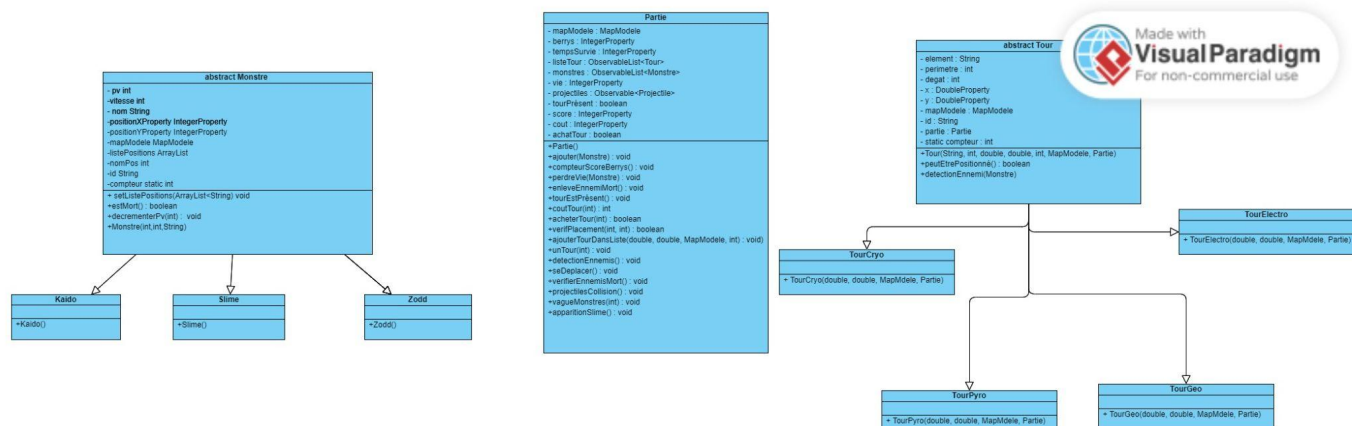
## 2. Diagrammes de classe (10 points pour SAE S2.01)

1 : Vous donnerez une vision d'ensemble de la partie modèle de votre programme à l'aide d'un ou de plusieurs diagrammes de classe commentés.

2 : Vous choisirez des parties du modèle que vous considérez particulièrement intéressantes du point de vue de la programmation à objets (héritage, composition, polymorphisme...).

Vous expliquerez vos choix et les illustrerez par des diagrammes de classe.

Chaque diagramme utilisé doit être focalisé sur un objectif de communication. Il ne doit par exemple pas forcément montrer toutes les méthodes et dépendances, mais juste ce qui est nécessaire pour montrer ce que le diagramme veut montrer. Chaque diagramme doit être commenté.



Nous avons choisi de vous présenter les diagrammes de classes :

- Nous avons décidé de ne pas préciser les get et set car les diagrammes de class serait bien trop long si nous les avons ajouté
- Monstre et Tour et ses sous classes : car ce sont des éléments clés du tower defense,
- Nous avons décidé de mettre monstre et tour en abstract class, pour empêcher la création de nouveau monstre ou de nouvelle tour non répertorié
- Nous avons décidé de créer le diagramme de classe de Partie car c'est dans cette class que la majorité des méthodes liant le joueur et le jeu sont.
- Nous avons décidé de faire le diagramme de la class MapModele car elle est très importante, c'est ici qu'on crée la map

### 3. Tests JUnit (15 points pour SAE S2.01)

1 ou 2 classes bien choisies devront être accompagnées de tests JUnit cohérents et raisonnablement complets.

Les tests sont disponibles dans le main de notre git.

### 4. SAE S2.02

#### Structures de données (30 points)

##### Classe Partie :

ObservableList<Monstre> monstres : elle est utilisée afin que l'on puisse supprimer les ennemis une fois qu'ils arrivent au Royaume ou sont éliminés par les projectiles.

ObservableList<Tour> tours : permet d'ajouter ou d'enlever les tours du jeu.

ObservableList<Projectile> projectiles : permet aux projectiles d'apparaître dès que la tour a détecté l'ennemi ou bien soit éliminés lorsqu'ils ont attaqué l'ennemi.

##### Classe Projectile et Tuile :

Les projectiles et les tuiles possèdent des property pour leur position.

#### Algorithmique (70 points)

##### Déplacement Monstre :

L'algorithme le plus intéressant et potentiellement le plus difficile est l'algorithme de déplacement qui utilise 3 classes (mapModele, Tuile et Monstre)

La classe MapModele

(dans la classe MapModele, la méthode tuileSuivante est intéressante, elle retourne une Tuile (une image de 32px \* 32 px) et passe à la tuile suivante (elle ne suit que les 1 de la MapModele et affiche un message de où se situe la tuile 1 suivante (en haut, en bas, à gauche, à droite) ) )

Cette méthode permet de retourner une ArrayList avec les positions à suivre.

La classe Monstre

Dans la classe Monstre la méthode setListePositions qui reçoit une ArrayList de liste de Directions qui est initialisé dans le constructeur à getListeDirection() (qui utilise la méthode tuileSuivante pour retourner en String une direction). Cette méthode ne retourne rien, c'est un void et c'est ce qui va permettre de déplacer un monstre

##### PlacerTour :

Un autre algorithme intéressant est l'algorithme de placement de la tour  
Il utilise 3 classes le controller( package Controller),la classe Tour (package modele.lestours)  
et la classe partie

La classe Controller

Dans un premier temps on commence par obtenir les coordonnées de la souris, une fois cela si le joueur clique sur une tour, on vérifie si le joueur a assez d'argent pour acheter une tour. S'il a assez d'argent il clique sur le panneau de jeu et on vérifie s'il est bien sur un emplacement de tour grâce à la méthode autorisationPlacement(dans le modèle) s'il a l'autorisation de placer une tour et qu'aucune autre tour est placée sur cet emplacement alors il pourra la positionner, l'image se positionnera au centre de la Tuile.

### **seDeplacer Projectile :**

Il y a également un algorithme qui permet au projectile de suivre les ennemis. Pour cela on a besoin des coordonnées du monstre et de savoir si le monstre est détecté par la Tour S'il est détecté alors des projectiles sont créés dans la classe Tour et les suivent.

## **5. Document utilisateur (40 points pour SAE S2.05)**

- description du jeu (son objectif, sont univers ...)
- Présentation du jeu avec captures d'écran.
- Mécanique des actions utilisateurs
- Caractéristiques des entités du jeu (les tours et les ennemis) et leurs dépendances éventuelles.
- Toute fonctionnalité définie et utilisable doit être documentée. A l'inverse, ne pas documenter les fonctionnalités non encore utilisables.

**Description du jeu** : Notre jeu est un *tower defense* sur un thème fantastique.

*Le royaume d'Arrassia*, l'un des royaumes les plus riches et puissants, mais aussi le plus isolé du continent de Celestia, qui *se fait envahir par des monstres* qui tentent d'assouvir leurs besoins primaires en pillant, massacrant et en détruisant tout sur leurs passage.

D'après les informations récolté par de courageux éclaireurs, nous savons que 3 races de monstres se sont alliés ; les *Slimes*, les *Zodds* et les *Kaidos*, et qu'une hiérarchie s'est établie selon la loi du plus fort : les Kaidos commandent les Zodds qui commandent les Slimes.

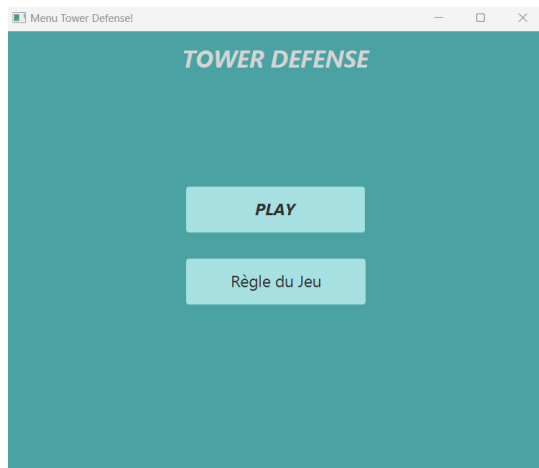
Pour se défendre, Arrassia utilise leurs nouvelles technologies, les tours magiques élémentaires.

Dans ce jeu votre but est de *protéger* le royaume d'Arrassia en *achetant des tours* élémentaires avec la monnaie local, les *Berrys*, et de les placer stratégiquement, car une fois placé, pas de retour en arrière.

Comment se procurer des berrys ? En éliminant des monstres !

**Présentation du jeu + Mécanique des actions utilisateur :**

Voici la page d'accueil de notre jeu :



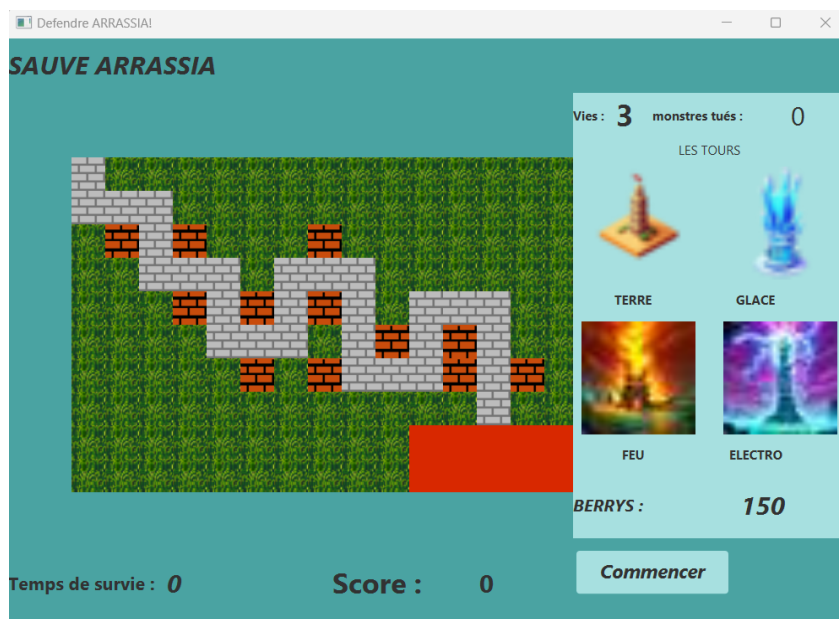
Voici la présentation du lore de notre jeu :



- Lorsque vous cliquez sur le bouton "TOURS ELEMENTAIRES" vous y verrez les caractéristiques des tours (que je vais expliquer prochainement).
- Lorsque vous appuyez sur "Continent de Celestia" vous verrez la map de Celestia :



Et lorsque vous cliquez sur “DEFENDRE LE ROYAUME !” le jeu commence : Vous pouvez commencer à placer les tours élémentaires pour défendre éliminer les ennemis



Au centre se trouve la map :

- les briques marron sont les espaces réservé aux tours élémentaires
- l'herbes verte est juste pour décorer
- le chemin gris qui commence en haut à gauche est le chemin emprunter par les montres pour se rendre au royaume d'Arrassia
- Arrassia qui est représenté par un rectangle rouge en bas à droite

A droite se trouve le menu qui vous permet d'acheter et placer des tours : pour en acheter une, il vous faut remplir différentes conditions :

- il faut cliquer sur l'image de la tour souhaiter
- puis cliquer sur l'un des emplacement disponible

attention !

- si une tour à déjà été placé à l'endroit souhaité
- ou si vous n'avez pas assez de berrys
- vous ne pourrez pas placer de tours élémentaire

Vous pouvez annuler votre achat de tour (tant qu'elle n'a pas été placé) en cliquant sur une autre tour, et la tour choisie sera celle que se placera si toutes les conditions sont remplies.

Vous verrez en temps réel :

- votre temps de survie (en bas à gauche)
- votre score (en bas au centre)
- Vos vies restantes (en haut du menu), vous commencez avec 3 vies, à chaque fois qu'un monstre parvient au royaume, vous perdez une vie, et lorsque vous perdez toutes vos vies le jeu s'arrête et le message "vous avez perdu" apparaît dans le terminale.
- l'endroit où est inscrit "monstres tués" est en réalité le nombre de monstres présents sur le terrain (monstres à tuer).
- En dessous des tours se trouve le nombre de berrys dont vous disposez, celui-ci se met à jour lors de l'achat de tours et lorsqu'un monstre est éliminé.

Lorsque vous êtes prêt, cliquez sur le bouton "Commencer", et la première vague de monstres ainsi que le temps de survie commenceront.

Voici un visuel d'une partie en cour :



Voici une un visuel de lorsqu'on a plus de vies :



### Caractéristiques des entités du jeu (les tours et les ennemis) et leurs dépendances éventuelles :

Au royaume d' Arrassia, nous avons 4 tours élémentaires qui lancent des projectiles. Les tours ont respectivement un périmètre et un nombre de dégâts différents, données qui seront utilisées par les projectiles :

Caractéristiques	TourGeo	TourCryo	TourPyro	TourElectro
périmètre	96px / 3 tuiles	64px / 2 tuiles	48px / 1.5 tuiles	32px / 1 tuile
dégâts	100	175	225	300

3 races d'ennemis tentent d'envahir le royaume :

- Les *slimes*, se sont des ennemis bleus, peu résistants mais cette faible résistance est compensée par leur rapidité. Sachant qu'ils sont plutôt faible ils rapportent un faible butin
- Les *Zodds* sont des ennemis beaucoup plus résistants que les slimes, ils sont donc légèrement moins rapides qu'eux (les slimes). Les Zodds plus résistants que les slimes rapportent un butin plus important que les slimes.
- Les *Kaidos* se rapprochent le plus des boss finaux. Ils sont extrêmement résistants rapporte un butin important et ont la même vitesse que les Zodds.

Caractéristiques	Slime	Zodd	Kaido
pv	350	1000	3500
vitesse (en px)	5	2	2
butin (en berry)	5	15	45
score	15	50	100

Il y a différentes vagues qui s'alternent en fonction du temps.

Les ennemis apparaissent en fonction du temps par vague, c'est-à-dire :

pendant 30 secondes, il va y avoir des slimes qui apparaîtront toutes les 3s ou 2s, selon les vagues.

Entre chaque changement de vague il y a un léger temps de pause.

Les slimes apparaissent dès le début et continuent d'apparaître tout au long des vagues

Le premier Zodd apparaît à la 85e secondes

Le premier Kaido apparaît à la 260e secondes

Les vagues ne prennent pas en compte s'il y a des ennemis présents sur le terrain comme dit précédemment le temps est maître de l'apparition des monstres.

Les ennemis disparaissent lorsque leurs pv atteignent 0 /lorsqu'ils meurent (pv = 0).

Jouez à ce jeu avec vos amis et comparez vos scores !