



**UNIVERSITÉ ÉVRY**  
**PARIS-SACLAY**

# Projet : Tower Defense

Université Évry Paris-Saclay  
L3 MAGE FI  
2025-2026

20231925 NARAYANASSAMYCHETTIAR VISWADEVI  
20245194 MEZEGHRANE ZINEDDINE  
20233004 PLUVINET-VIAENE ARMAND  
20230692 NGUARA NGOMA MARC

## Table des matières

1.INTRODUCTION .....	3
1.1 Description du jeu .....	3
1.2 Rôles .....	3
1.3 Planning et affectations.....	3
2. MODELISATION.....	5
2.1 Diagramme de cas d'utilisation .....	5
2.2 Diagramme d'activité .....	5
2.3 Diagramme de séquence.....	6
2.4 Diagramme de classe.....	8

# 1.INTRODUCTION

## 1.1 Description du jeu

Ce projet propose de réaliser une version originale d'un Tower Defense asymétrique en mode un contre un. Un joueur incarne l'Attaquant en envoyant des vagues de créatures, tandis que l'autre joue le rôle du Défenseur en plaçant et améliorant des tours. Le jeu se repose sur un système de tours alternés comprenant des phases distinctes de préparation et d'attaque. La partie se déroule sur un plateau fixe reliant les bases des deux joueurs et se termine par la victoire de l'Attaquant ou du Défenseur selon les conditions de fin de jeu.

## 1.2 Rôles

- **1 responsable conception & cheffe de projet** (Viswadevi)  
Conception globale du projet (UML, architecture), rédaction de la documentation, organisation et coordination du groupe (répartition des tâches, réunions, planning, communication), suivi de l'avancement.  
Participation ponctuelle au développement : aide sur certaines parties du code, compréhension et relecture des implémentations.
- **3 développeurs** (Armand, Marc, Zineddine)  
Implémentation des fonctionnalités principales, choix techniques, structure du projet, gestion du Git et des tests.  
Participation à la conception et à la validation du projet : relecture et validation des diagrammes UML, contribution aux décisions de conception et à la gestion de projet.

## 1.3 Planning et affectations

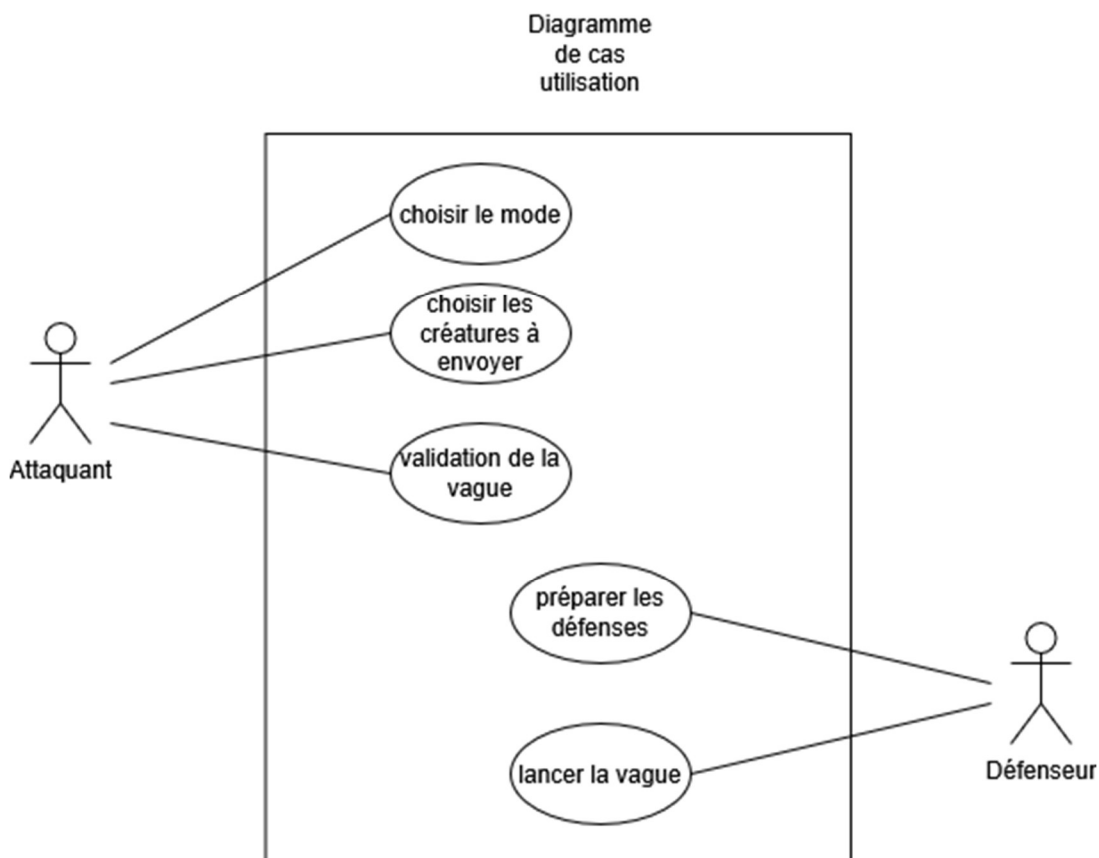
Semaines	Tâches	Affectations
1 à 5	<ul style="list-style-type: none"><li>• Analyse du sujet</li><li>• Modélisation (UML)</li><li>• Planning + répartition des tâches</li><li>• Recherche technique (outils, librairies Python, architecture)</li><li>• Premier rendu : modélisation + planning + affectation</li></ul>	<ul style="list-style-type: none"><li>• Tous</li><li>• Viswa</li><li>• Viswa</li><li>• Tous</li><li>• Viswa</li></ul>
6	<ul style="list-style-type: none"><li>• Création et organisation du dépôt Git</li><li>• Mise en place des branches</li><li>• Finalisation des diagrammes UML (classes, séquence, activité).</li><li>• Répartition définitive des tâches et planification interne.</li></ul>	<ul style="list-style-type: none"><li>• Tous</li><li>• Tous</li><li>• Viswa</li><li>• Tous</li></ul>

7	<ul style="list-style-type: none"> <li>• Plateau de jeu (dimensions, zones de placement, bases attaquant/défenseur)</li> <li>• Chemin (fixe ou aléatoire)</li> <li>• Début des classes</li> <li>• Affichage du plateau</li> <li>• Visualisation du chemin</li> </ul>	<ul style="list-style-type: none"> <li>• Armand, Marc, Zineddine</li> <li>• Armand, Marc, Zineddine</li> <li>• Marc, Armand, Viswa</li> <li>• Armand, Viswa</li> <li>• Armand, Marc</li> </ul>
8	<ul style="list-style-type: none"> <li>• Phase de préparation</li> <li>• Phase d'attaque</li> <li>• Alternance des tours</li> <li>• Récompenses liées aux créatures détruites</li> <li>• Synchronisation des phases</li> <li>• Affichage des ressources</li> </ul>	<ul style="list-style-type: none"> <li>• Marc</li> <li>• Marc</li> <li>• Zineddine, Armand</li> <li>• Zineddine, Armand</li> <li>• Armand, Marc, Zineddine</li> <li>• Armand, Marc</li> </ul>
9	<ul style="list-style-type: none"> <li>• Logique d'envoi des vagues</li> <li>• Caractéristique des créatures</li> <li>• Capacités des créatures (invisibilité, mort, ...)</li> <li>• Interface composition vague</li> </ul>	<ul style="list-style-type: none"> <li>• Marc, Armand, Viswa</li> <li>• Tous</li> <li>• Tous</li> <li>• Armand, Marc</li> </ul>
10	<ul style="list-style-type: none"> <li>• Tours de dégâts</li> <li>• Tours de contrôle</li> <li>• Interaction tous/créature</li> <li>• Portée, ciblage</li> <li>• Niveaux de tours</li> <li>• Coût</li> <li>• Placement des tours</li> <li>• Interface</li> </ul>	<ul style="list-style-type: none"> <li>• Armand, Marc</li> <li>• Armand</li> <li>• Armand, Marc</li> <li>• Armand, Marc, Zineddine</li> <li>• Armand, Viswa</li> <li>• Armand, Viswa</li> <li>• Armand, Viswa</li> <li>• Armand, Marc</li> </ul>
11	<ul style="list-style-type: none"> <li>• Attaquant : Infos limitées sur les tours</li> <li>• Défenseur invisibilité des vagues avant le lancement</li> <li>• Brouillard</li> <li>• Génération de vague</li> <li>• (Sons de tours, créatures)</li> </ul>	<ul style="list-style-type: none"> <li>• Armand, Marc</li> <li>• Armand, Marc</li> <li>• Armand, Marc</li> <li>• Tous</li> <li>• Armand</li> </ul>
12	<ul style="list-style-type: none"> <li>• Finalisation du code et des derniers correctifs.</li> <li>• Tests</li> <li>• Débogage</li> <li>• Optimisation des performances</li> </ul>	TOUS
13	<ul style="list-style-type: none"> <li>• Correction derniers bugs</li> <li>• Optimisation du code</li> <li>• Nettoyage du dépôt Git</li> </ul>	TOUS
14	<ul style="list-style-type: none"> <li>• Vérification structure du projet</li> <li>• Préparation individuelle soutenance</li> </ul>	TOUS

## 2. MODELISATION

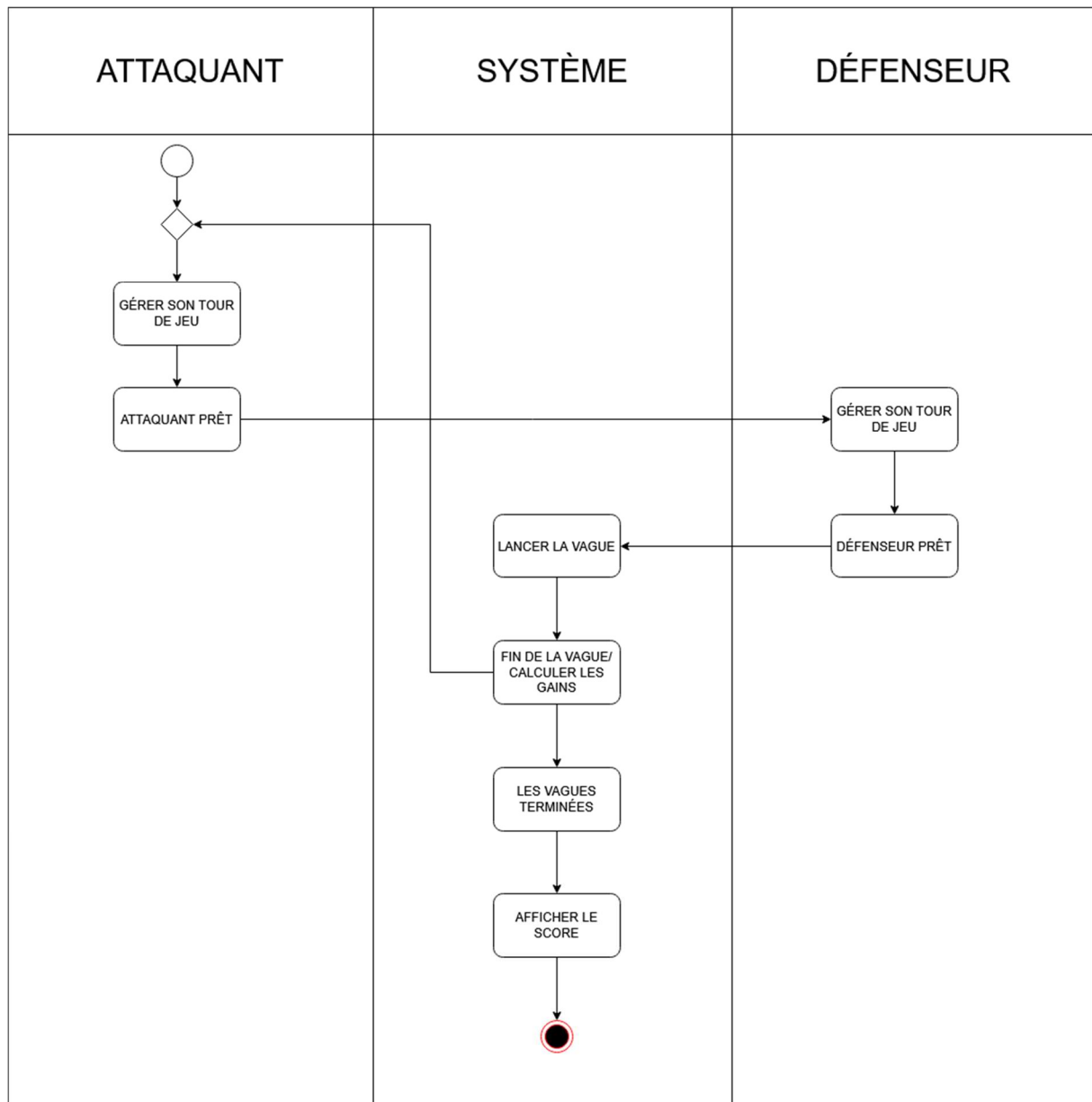
### 2.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation présente l'ensemble des actions que peuvent réaliser l'attaquant et le défenseur au cours du jeu. L'attaquant intervient en premier : il choisit le mode de jeu, sélectionne les créatures à envoyer puis valide la vague. Une fois cette étape terminée, le défenseur prépare ses défenses et lance la vague.



### 2.2 Diagramme d'activité

Ce diagramme d'activité décrit le déroulement complet d'un tour de jeu en montrant comment l'attaquant, le défenseur et le système interagissent. L'attaquant et le défenseur commencent chacun par gérer leur tour, puis indiquent qu'ils sont prêts. Une fois ces deux conditions réunies, le système lance la vague, en calcule les gains à la fin, vérifie si toutes les vagues ont été jouées, puis affiche le score final.



## 2.3 Diagramme de séquence

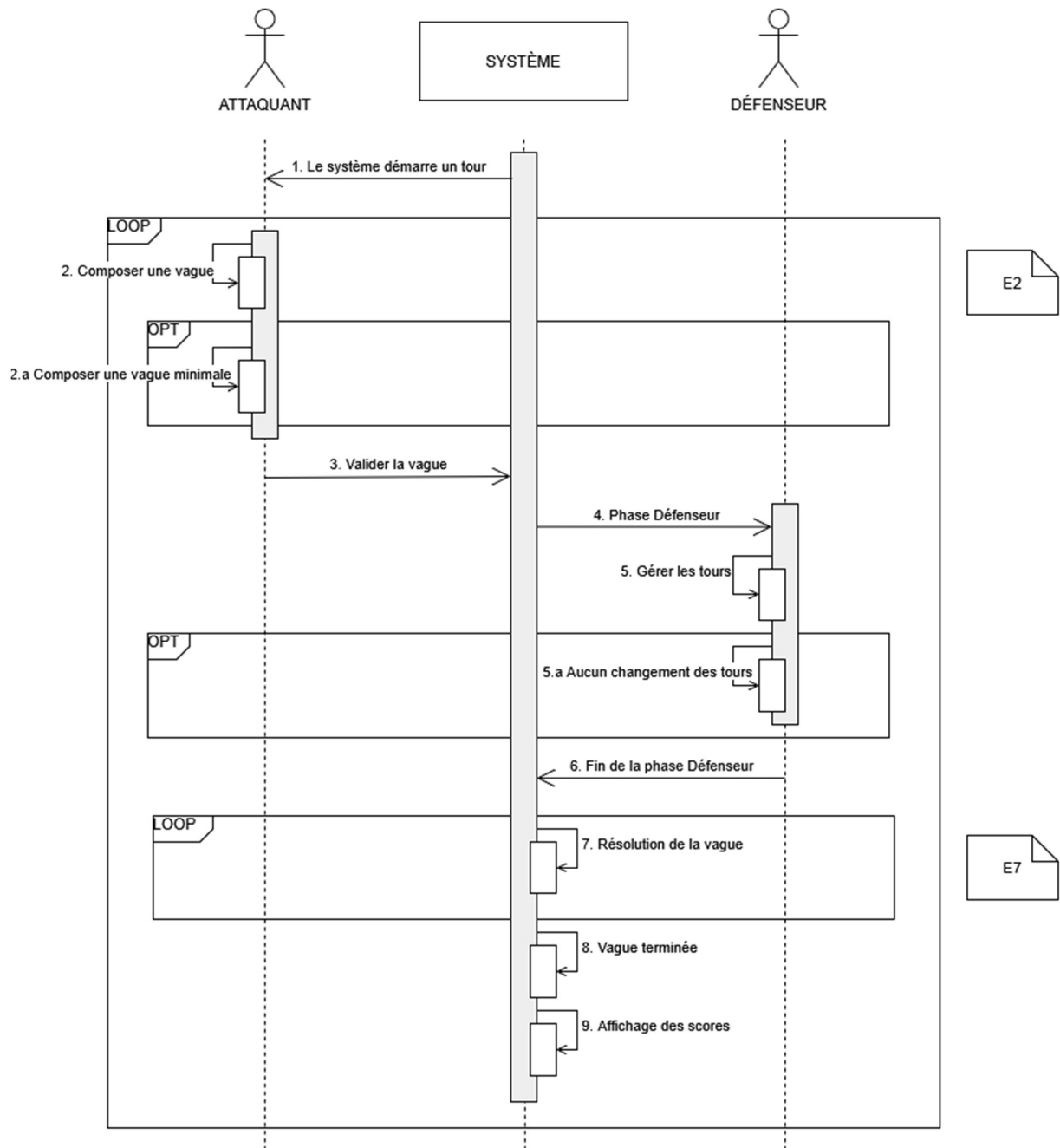
### CAS NOMINAL

Le cas nominal définit le chemin standard suivi par les acteurs pour accomplir une action sans interruption. Il décrit la succession logique des étapes réalisées par l'Attaquant, le Défenseur et le système, depuis le début du tour jusqu'à la résolution de la vague et l'affichage du résultat final.

NOM	TOWER DEFENSE
ACTEURS	Attaquant, Système de jeu, Défenseur
DESCRIPTION	Un tour pendant lequel l'Attaquant prépare et lance une vague de créatures ensuite le Défenseur gère ses tours. Les vagues se répètent jusqu'à ce que la base du Défenseur tombe à 0 PV (victoire de l'Attaquant) ou que l'Attaquant n'ait plus assez de monnaie pour

	lancer une nouvelle vague (victoire du Défenseur). Le système affiche ensuite le résultat.
PRÉCONDITIONS	<ul style="list-style-type: none"> <li>• La partie est initialisée (plateau, chemin, bases, ressources de départ).</li> <li>• L'Attaquant dispose d'au moins une certaine quantité d'or au début du tour.</li> <li>• La base du Défenseur possède des PV &gt; 0.</li> </ul>
SCÉNARIO NOMINAL	<ol style="list-style-type: none"> <li>1. Le système démarre un nouveau tour et donne la main à l'Attaquant.</li> <li>2. L'Attaquant consulte sa monnaie et compose une vague dans la limite de son budget.</li> <li>3. L'Attaquant valide la vague.</li> <li>4. Le système passe en phase Défenseur.</li> <li>5. Le Défenseur place, améliore ou vend des tours selon sa monnaie disponible.</li> <li>6. Le Défenseur termine sa phase.</li> <li>7. Le système lance la résolution de la vague. (Les créatures apparaissent au début du chemin et avancent automatiquement. Les tours attaquent automatiquement les créatures à portée. Les créatures détruites sont retirées et génèrent des gains. Les créatures arrivant à la base font perdre des points de vie au Défenseur.)</li> <li>8. La vague se termine lorsque toutes les créatures sont détruites ou arrivées.</li> <li>9. Le système affiche le résultat.</li> </ol>
SCENARIO ALTERNATIFS	<p>2.a. L'Attaquant n'a pas assez de monnaie pour une vague et compose une vague minimale très faible.</p> <p>5.a Le Défenseur ne place ni n'améliore aucune tour (manque de monnaie ou choix stratégique).</p>
ENCHAINEMENTS D'ERREUR	<p>E2 L'Attaquant ne dispose d'aucune monnaie pour composer une vague. Le système déclare immédiatement la victoire du Défenseur et la partie se termine.</p> <p>E7 Les points de vie de la base du Défenseur atteignent 0 durant la résolution. Le système interrompt le tour et déclare la victoire de l'Attaquant.</p>
POSTCONDITION	<ul style="list-style-type: none"> <li>- La partie est terminée (victoire de l'Attaquant ou du Défenseur),</li> <li>- L'état du jeu est mis à jour et un nouveau tour peut commencer.</li> </ul>

Le diagramme de séquence illustre le déroulement d'un tour de jeu en détaillant les échanges entre l'Attaquant, le Système et le Défenseur. Il montre l'ordre chronologique des actions et des messages transmis entre les trois acteurs, depuis le début du tour jusqu'à l'affichage final des scores.



## 2.4 Diagramme de classe

Le diagramme de classe présente l'architecture principale du jeu de Tower Defense en mettant en évidence les entités du système, leurs attributs, leurs méthodes et leurs relations.

On y trouve d'abord les classes liées à la gestion du tour et de la visibilité, comme VisionSystem et TurnPlayer, qui coordonnent respectivement les informations visibles pour chaque joueur et le déroulement des phases de jeu. Le GameBoard structure le plateau grâce à ses dimensions et à la composition de Cell, tandis que les Path et Waypoint définissent le trajet emprunté par les créatures.

Les classes Tower et Creature représentent les éléments centraux du gameplay : les tours possèdent des caractéristiques d'attaque et les créatures disposent des caractéristiques de déplacement et de survie. La classe Effect permet d'appliquer des bonus ou malus sur ces créatures.



La classe Player sert de base aux deux rôles du jeu : Attacker et Defender. L'Attaquant peut composer et lancer des vagues via la classe Wave, tandis que le Défenseur peut placer ou améliorer des tours. Les relations entre classes montrent notamment que les tours attaquent les créatures, que les créatures suivent un chemin, et que les vagues regroupent plusieurs créatures et effets.

