

# RAPPORT DE PROJET POO

Minh Nhut NGUYEN  
Massyl ADJAL  
Manel MOULAHCENE

## 1. Présentation du projet et de son contexte :



Ce projet de jeu en 2D, développé en Python en utilisant la bibliothèque Pygame, propose une expérience immersive avec des fonctionnalités riches telles que des personnages uniques, des compétences spéciales, et des environnements variés. Le projet repose sur deux modules principaux, *game.py* et *unit.py*, qui gèrent respectivement la logique globale du jeu et les caractéristiques des unités.

## 2. Répartition du travail dans l'équipe :

tache/membre	Importance tache	Massyl	Manel	Minh
Creation de la map	Tres important			R
initialisation de la partie	Tres important			
Dynamisme du tour par tour	Tres important	R		
Controles et gameplay	Tres important	R		
Limite de mouvement et endurance	Tres important			
Execution de la compétence	Tres important			
Dynamique de visibilité et déplacement	Bonus			
Creation des champions	Tres important		R	
Déplacement des champions	Tres important		R	
compétences des champions	Tres important			R
caractéristiques des champions	Tres important			R
Add sound design	Bonus			

Nous nous sommes réunis pour définir la répartition du travail la plus efficace possible. À l'aide de Google Sheets, nous avons recensé les premières étapes à suivre après avoir pris connaissance du contenu du projet. Chaque membre de l'équipe a été assigné à 2 à 3 missions distinctes, en plus de créer les personnages et les effets qui composent le jeu. Une fois un certain nombre de tâches réalisées, nous avons pu visualiser l'ensemble du jeu et avons décidé que l'étape suivante consistait à identifier les exigences essentielles à inclure dans le rapport, afin de maximiser la valorisation des points. Enfin, nous avons collaboré sur le code et le rapport pour atteindre les résultats escomptés.

## 3. Présentation des fonctionnalités de base implémentées dans le jeu :

- Implémentation de différentes unités : Nous avons conçu trois types d'unités distinctes héritant de la classe mère **Unit**, chacune représente un type unique : **Sorceress**, **Samurai** et **Shinobi** caractérisés par :
  - Un nombre de points de vie (ici une barre de vie a été rajoutée au niveau des unités des personnages)
  - Une statistique d'endurance (1 point d'endurance représente un bloc de mouvement pour l'unité)
  - Une statistique d'attaque
  - Une statistique de défense (en pourcentage de 0 à 100)
- Implémentation de **9 différentes compétences** avec animations et effets sonores héritant de la classe mère **Skill**, certaines compétences ont des effets de zone, d'autre cible une seule unité.. (chaque personnage possède 3 compétences différentes)
- Implémentation de différentes cases avec leurs effets distinctes sur le gameplay (**8 terrains différents** : grass, snow, water, magma, apple\_tree, mud, wall, bush)
- Implémentation du déplacement des unités en temps réel et limité grâce à **l'endurance**.
- Implémentation de la portée des compétences.
- Implémentation de la zone d'effet des compétences.
- Implémentation du calcul des dégâts (Nous avons appliqué une formule de dégât en utilisant la puissance du personnage et la défense de la cible)

## 4. Présentation des fonctionnalités supplémentaires implémentées dans le jeu :

- A. **L'écran Menu** pour choisir l'option de jeu : Solo DeathMatch + Multi Player DeathMatch, avec musique de fond.
- B. L'écran de sélection de carte en utilisant les fichiers csv + bouton de retour à l'écran Menu + Titre
- C. L'écran de sélection de personnages + Titre
- D. L'ajout de la musique et des effets sonores pour dynamiser l'expérience de jeu
- E. L'ajout de l'animation des compétences
- F. Une partie d'IA de gestion des mouvements et des attaques des ennemies
- G. L'ajout d'une section à l'écran dédiée à l'affichage des informations, comprenant la barre de points de vie, la barre d'endurance, les icônes des trois compétences avec leurs descriptions, ainsi que les indications pour activer les compétences à l'aide des touches directionnelles et de la souris.
- H. L'ajout de 6 compétences (9 compétence en total) :
  - 1. "Yennefer" - "Sorceress" - Skills :
    - a. Purple Chaos
    - b. Poison Apocalypse
    - c. Healing
  - 2. "Shogun" - "Samurai" - Skills :
    - i. Ichimonji
    - ii. Sky Clear
    - iii. Samurai Grave
  - 3. "Sekiro" - "Shinobi" - Skills :
    - i. Shuriken
    - ii. Death Shadow
    - iii. Shadow Berserk

## 5. Justification des choix de conception dans le diagramme de classes :

Héritage des Classes Sorceress, Samurai et Shinobi depuis la classe Unit :

Ces trois classes représentent les différents types d'unités, donc une grande partie des attributs et méthodes de ces classes sont les mêmes (comme health, move(), attack()...), il est donc très utile d'utiliser l'héritage et ainsi définir ces attributs et méthodes communes une seule fois dans la classe Unit.

Héritage des Classes représentants les différents Skills et la classe abstraite Skill :

Ceci a un rôle dans l'explicabilité du code, c'est pour bien montrer que les 9 classes Ichimonji, Purple\_Chaos, Healer, Sky\_Clear,

Samurai\_Grave, Poison\_Master, Assassin\_Flicker, Shuriken, Shadow\_Berserk representent bien des skills.

## Composition entre Les Skills et les Unités :

Les skills sont propres à chaque unité et c'est l'unité qui utilise un skill, il est donc naturel d'utiliser une relation de composition ou les skills sont instanciés par l'unité pour ensuite être utilisés dans le jeu par l'unité en question.

## Aggregation entre les 3 types d'unités et Game :

Les unités sont utilisées uniquement par l'intermédiaire de la classe Game, mais celles-ci sont instanciées en dehors pour définir un dictionnaire regroupant tous les personnages jouables, il s'agit donc d'une relation d'agrégation.

## 6. Analyse des défis rencontrés et solutions apportées :

Les difficultés rencontrées lors de notre projet étaient principalement liées à la gestion des conflits de code et à la coordination entre les membres de l'équipe, mais l'utilisation de Git a simplifié la gestion des versions et la collaboration sur le même code.