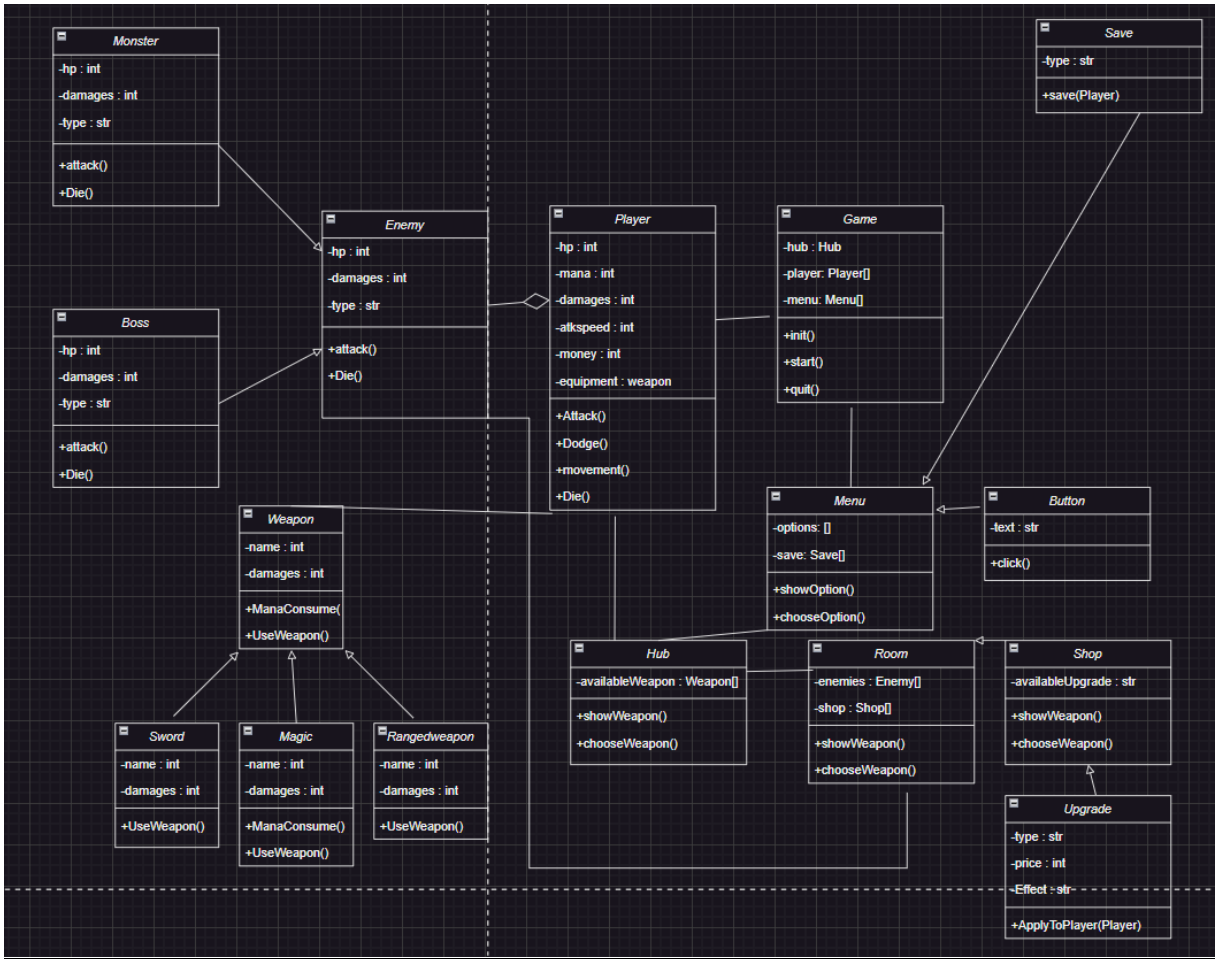
Hemeryck Quentin – Descamps Max – Blangenois Arthur – Olivier Florian
Groupe 15 :

Analyse du Projet

Diagramme de classe + explications



description : Le Jeu se lance avec la classe Game. Ensuite, ouverture du menu (classe Menu) avec ses boutons clickables (classe Button) et on y choisit le mode. Après, déplacement dans le hub (classe hub). Le player apparait avec ses 3 armes (class Weapon + sous classes Magic, Sword, RangedWeapon) Une option save est disponible dans le menu (classe Menu). Après le hub, implémentation de la classe "Room" qui est en composition avec la classe Enemy dont les héritiers sont Monster et Boss. La classe Shop hérite de la classe Room et ce shop hérite de la classe Upgrade

Analyses des besoins

- **Fonctionnels** :
    - Un personnage principal avec des capacités évolutives.
    - Système de statistiques permettant de voir le niveau de vie , attaque , vitesse de déplacement et vitesse d'attaque
    - Deux niveaux principaux avec une difficulté progressive : deux modes de jeux dont un mode classique et un mode contre la montre.

- Différents types d'ennemis avec des comportements variés : 2-3 ennemies et boss final lors de la fin de chaque round augmentant ainsi la difficulté.
- Système de récompenses et progression du joueur : le joueur aura la possibilité de gagner des pièces lors des différentes combats contre les ennemies .
- Collectibles : Objets à collecter dans le jeu pour débloquer des récompenses ou des bonus.
- Documentation complète : Fournir une documentation détaillée pour aider les joueurs à comprendre toutes les fonctionnalités et mécaniques du jeu : celle-ci se trouvera dans la fiche technique du jeu ce qui apportera une meilleure compréhension de l'utilisation du jeu.
- Système de sauvegarde : Cette dernière permet a l'utilisateur d'enregistrer les parties jouées

- **Non-fonctionnels** :
  - Graphismes et audio de haute qualité : Assurer une expérience visuelle et sonore immersive avec des graphismes détaillés et une bande sonore appropriée.
  - Interface utilisateur intuitive et réactive : possibilités de mettre pause durant la partie
  - Compatibilité avec une seule plateformes : ordinateurs

Répartition des taches du premier jour :

Hemeryck Quentin :

- Création du document reprenant l'analyse complète
- Création du dépôt distant Git + différentes branches assignées
- Résumé des taches à effectuer
- Création du menu de base
- Création d'un menu de pause

Descamps Max :

- Création de l'apparition aléatoire des mobs
- Création du diagramme de classe
- Création du Design de la première Map

Blangenois Arthur :

- Recherche des sprites à utiliser
- Commencement de la génération des montres aléatoire
- Première partie du diagramme de classe
- Création et mise en page du Word commun

Olivier Florian :

- Recherches des sprites
- Apprentissage title
- Recherche des différents sounds

Hemeryck Quentin – Descamps Max – Blangenois Arthur – Olivier Florian
Groupe 15 :

- Commencement des dialogues
- Elaboration des déplacements

Répartition des taches

Weapons / score / magie / shop => Olivier Florian

Personnages / Animation / Attaque => Blangenois Arthur

Ennemie / Boss/ Rooms => Descamps Max

Menu / Sauvegarde / Assistances Rooms => Hemeryck Quentin

Décomposition du jeu :

1. Choix des Armes

Le joueur doit choisir entre trois armes différentes, chacune offrant un style de combat unique :

- **La Lance Tomate**: : Le joueur lance des projectiles en forme de petit bébé Mario qui se déplacent rapidement vers les ennemis.
- **La Lance de Bébé Mario** : Le joueur attaque à distance, utilisant la portée pour abattre les ennemis avant qu'ils n'atteignent le personnage.
- **L'Épée de Feu** : Une arme pour une stratégie offensive et puissante, infligeant des dégâts élevés au corps à corps.

2. 1ère étage

le joueur entre dans la première salle où il doit combattre une vague d'ennemis. Les ennemis sont de difficulté modérée pour permettre au joueur de se familiariser avec les mécaniques de combat. Survivre à la première vague ouvre la porte de la deuxième salle, où le joueur affronte une deuxième vague plus difficile. Les ennemis ici sont plus résistants et nombreux, exigeant une meilleure maîtrise des compétences du joueur.

3. Le Shop

Après avoir vaincu les ennemis des deux premières salles, le joueur accède au Shop. Dans le Shop, les écus collectés sur les ennemis peuvent être échangés contre des améliorations :

- **Vie** : Augmente la santé maximale du joueur.
- **Dégâts** : Augmente les dégâts infligés par le joueur.
- **Vitesse d'Attaque** : Augmente la fréquence des attaques du joueur.

4. 2ème étage

Survivre à la première vague ouvre la porte de la deuxième salle, où le joueur affronte une deuxième vague plus difficile. Les ennemis ici sont plus résistants et nombreux, exigeant une meilleure maîtrise des compétences du joueur.

5. Le Shop

Après avoir vaincu les ennemis des deux premières salles du 1 ère étage, le joueur accède au Shop. Dans le Shop, les écus collectés sur les ennemis peuvent être échangés contre des améliorations :

- **Vie** : Augmente la santé maximale du joueur.
- **Dégâts** : Augmente les dégâts infligés par le joueur.
- **Vitesse d'Attaque** : Augmente la fréquence des attaques du joueur.

5. Combat contre le Boss

Le joueur doit ensuite affronter le Boss de l'étage 1 pour accéder à l'étage 2. Ce combat est plus intense et stratégique, nécessitant l'utilisation optimale des compétences et des améliorations du joueur. Vaincre le Boss est nécessaire pour progresser dans le jeu et donc avoir l'accès au second étage.

6. Deuxième Étage

Après avoir vaincu le Boss du 1ère étage, le joueur monte au deuxième étage, qui suit la même structure mais avec une difficulté accrue. Le deuxième étage comporte également deux salles de combat suivies d'un Shop et un Boss final.

7. Fin du Jeu

Le jeu se termine lorsque le joueur a réussi à surmonter tous les défis du deuxième étage. Le joueur est alors récompensé pour sa réussite et peut voir un écran de victoire.

Analyse des besoins du Client :

Après discussions avec le client la définition des besoins a été établit selon les critères suivants :

1. Types d'armes

- **Question** : Souhaitez-vous avoir des styles d'armes différents comme Arc, Épée, Magie ?
- **Réponse** : Le client préfère des armes originales et créatives, comme un lancer de pastèque, plutôt que des armes traditionnelles comme l'arc et l'épée.

2. Structure du jeu

- **Question** : Je vous propose pour ce jeu 2 étages avec entre 4 à 6 salles. Cela vous suffirait-t 'il ?
- **Réponse** : Le client trouve que 2 étages avec 4 à 6 salles chacun sont suffisants pour le jeu.

3. Améliorations de statistiques

- **Question** : Concernant les améliorations de statistiques du style Vitesse d'attaque, Vitesse de déplacement, augmentation de dégâts. Est-ce que cela vous conviendrait ?
- **Réponse** : Oui, le client souhaite inclure des améliorations de statistiques telles que la rapidité des projectiles, en plus de la vitesse d'attaque, vitesse de déplacement, et
- augmentation des dégâts.

4. Modes de jeu

- **Question** : Concernant les modes de jeu, est-ce qu'un mode Normal et un mode Hardcore (Only boss dans chaque salle, avec récompenses plus élevées…) vous conviendrait ?
- **Réponse** : En plus des modes Normal et Hardcore, le client propose un mode contre la montre où le joueur doit compléter un étage dans un temps imparti, sinon c'est Game Over. Un système de speedrun est également suggéré.

5. Diagrammes de classes et séquences

- **Question** : Quels diagrammes réaliser ? Est-ce qu'on a besoin de faire les diagrammes de séquences, cas d'utilisation ?
- **Réponse** : L'objectif principal est de fournir un diagramme de classes clair et précis, montrant la structure du projet. Les diagrammes de séquences ne sont nécessaires que si elles sont cruciales pour expliquer la logique du projet.

Hemeryck Quentin – Descamps Max – Blangenois Arthur – Olivier Florian
Groupe 15 :

Résumé des besoins du client

- **Armes** : Originales et créatives, pas seulement traditionnelles.
- **Structure du jeu** : 2 étages avec 4 à 6 salles chacun.
- **Améliorations de statistiques** : Inclure vitesse d'attaque, vitesse de déplacement, augmentation des dégâts, et rapidité des projectiles.
- **Modes de jeu** : Mode Normal (boss dans chaque salle, récompenses élevées), et mode contre la montre (speedrun).
- **Diagrammes** : Diagramme de classes obligatoire, diagrammes de séquences optionnels mais utiles si nécessaires pour la compréhension du projet.

Speech Final

Concernant la généralité de la conception du jeu, Ce dernier doit offrir un personnage principal évolutif, des statistiques suivant santé, attaque, et vitesse, ainsi que deux niveaux avec une progression de difficulté. Il devrait inclure différents ennemis et un système de récompenses. Les besoins non-fonctionnels incluent des graphismes de qualité, une interface intuitive, et une compatibilité avec les ordinateurs. Les tâches sont réparties entre les membres de l'équipe pour couvrir tous les aspects du jeu, de la création des armes à la mécanique de jeu.

Explication des méthodologies

Pour structurer notre travail de groupe de manière efficace, nous utilisons les méthodes QQOQCP et le cycle V.:

Qui ?

Équipe de Développement : Notre groupe de quatre personnes :

- Hemeryck Quentin
- Descamps Max
- Blangenois Arthur
- Olivier Florian

Quoi ?

- Objectif : Réaliser un jeu vidéo en Python avec PyGame, inspiré du jeu HADES.
- Fonctionnalités :
- Un personnage principal avec des capacités évolutives
- Système de statistiques (santé, attaque, vitesse de déplacement et d’attaque)
- Deux niveaux avec plusieurs salles et ennemis variés
- Système de récompenses et améliorations
- Différents modes de jeu (normal, contre la montre)
- Graphismes et sons de haute qualité

Hemeryck Quentin – Descamps Max – Blangenois Arthur – Olivier Florian
Groupe 15 :

Où ?

- Lieu de Travail : Collaboration en ligne via des plateformes comme GitHub pour le code, Google Docs pour la documentation, et Discord pour la communication en temps réel.

Quand ?

- Semaine de projet du 13 au 17 mai 2024

Chronologie :

Jour 1: Création du dépôt Git, analyse des besoins, début de la conception des classes et du menu

Jour 2-3-4: Développement des fonctionnalités principales, création des niveaux et des ennemis, intégration des graphismes et sons

Jour 5: Présentation du jeu vidéo et explications des fonctionnalités de ce dernier.

Comment ?

- Méthodologie :
    - Utilisation de PyGame pour le développement
    - Répartition des tâches par spécialité (graphismes, sons, codage des mécaniques de jeu, conception des niveaux)
    - Réunions quotidiennes pour discuter de l'avancement et des obstacles et remise en question

Pourquoi ?

- Motivation : Apprendre et appliquer les concepts de développement de jeux vidéo, travailler en équipe, et développer des compétences en programmation avec Python et PyGame.

Cycle V (Vérification et Validation)

- Spécification : Définir clairement les besoins du jeu en termes de fonctionnalités et de performances.
- Conception : Créer des diagrammes de classes et des séquences pour structurer le code.
- Développement : Écrire le code en suivant les spécifications et la conception.
- Intégration : Combiner les différentes parties du code, intégrer les graphismes et les sons.
- Validation : Tester le jeu pour s'assurer qu'il répond aux exigences et corriger les bugs.
- Maintenance : Documenter le code et préparer des mises à jour