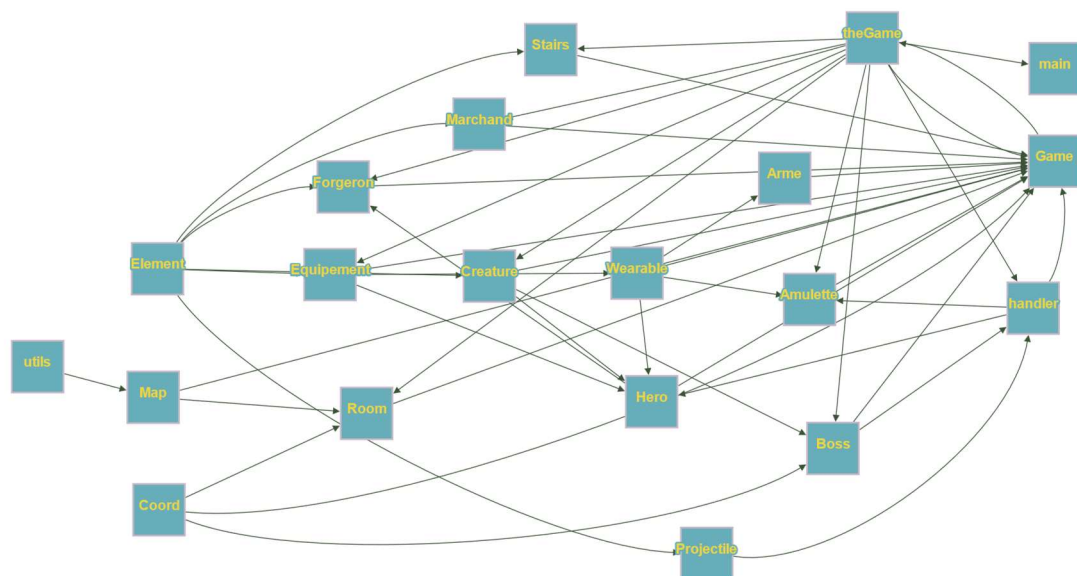


RAPPORT ROGUE-US :

Membres : Clement Andy / Martini Alexis / Pantani Timothé / Dufraigne Titouan

Bienvenue dans notre jeu, **ROGUE-US** ! Avant de commencer veuillez lire le READ ME.

Lien vers l'organigramme : <http://graphonline.ru/fr/?graph=ICkvfVBkdeEHqwtvZZcst>



Organigramme du code

Classe et leur méthode :

Amulette :

- regeneration()
- coupCrit()

Boss :

- spawnArmy()
- tpBoss()

Coord :

- distance()
- direction()

Creature :

- description()
- meet()

Element :

- description()
- meet()

Equipement :

- meet()
- use()

Forgerons :

- meet()

Game :

- buildFloor()
- buildBoss()
- buildStairs()
- addMessage()
- readMessages()
- randElement()
- randEquipement()
- randMonster()
- select()
- pause()
- drop()
- selectPerso()
- restart()
- play()

Handler :

- heal()
- teleport()
- tir()
- boost()
- nerd()
- revive()
- fullBoost()
- amasser()

Hero :

- description()
- fullDescription()
- checkEquipement()
- take()
- use()
- unEquip()
- inventSize()

- shot()
- throw()

Map :

- addRoom()
- findRoom()
- intersectNone()
- dig()
- corridor()
- reach()
- reachAllRooms()
- randRoom()
- generateRooms()
- checkElement()
- put()
- get()
- pos()
- rm()
- move()
- moveAllMonsters()
- vision()

Marchand :

- meet()

Room :

- intersect()
- center()
- randCoord()
- randEmptyCoord()
- decorate()

Stairs :

- meet()

TheGame :

- theGame()

utils :

- sign()

Wearable :

Modifications présentes dans la liste des éléments suggérer :

Gameplay

Elément de Gameplay. III (Inventaire limité) :

- Hero.size : Argument d'instance qui stocke la taille de l'inventaire.
- Hero.inventSize() : Vérifie si l'inventaire est plein. Si le nombre d'éléments dans l'inventaire est inférieur à la limite, renvoie True.
- Hero.take() : (Modification). Ne prend pas d'élément si l'inventaire est plein.
- Hero.description() : (Modification). Ajout d'un compteur d'item.

Elément de Gameplay. IV (Déplacement semi-intelligent) :

- Map.moveAllMonsters() : les créatures peuvent éviter un obstacle sur leur chemin afin d'atteindre le héros.

Elément de Gameplay. VI (Nuage de visibilité+) :

- Hero.vision : variable correspondant à la visibilité du héros
- Map.vision() : méthode copiant la map et retournant son équivalent avec du brouillard
- Map.fog (#) : le brouillard cachant la map

Objet

Elément d'Objet. II (Armes et gestion) :

- Wearable.__init__() : Gestion de l'emplacement des équipements
- Creation de la classe Arme (hérite de Wearable) : Permet de différencier les consommables avec les armes
- Hero.body : (argument d'instance) liste qui représente le corps du héros et permet de gérer les équipements que le héros a équipé. Chaque indice représente une partie du corps
- Hero.use() : (Modification). Distinction entre les éléments consommable et ceux qui peuvent s'équiper (équipement). Les consommables sont utilisés et les équipements sont ajoutés dans body
- Hero.unEquip() : Permet de déséquiper un équipement de body. L'équipement retourne dans l'inventaire.
- Handler.boost() : Permet d'augmenter les stats du héros (Quand on équipe un élément)
- Handler.nerf() : Permet de réduire les stats du héros (Quand on déséquipe un équipement par exemple)

Elément d'Objet. III (Armes de Jet) :

- Hero.shot() : Vérifie si le joueur possède une arme de distance équipée. Si oui il tire.
- Hero.throw() : Demande à l'utilisateur quand quelle direction il veut tirer.
- Handler.tir() : Sert à tirer une flèche dans une direction donnée. Elle peut être utilisée par le héros ou les créatures et peut donner des effets comme poison ou gel à la cible (avec une probabilité).

- Création de la Classe Projectile qui hérite de Elément. Il peut avoir une portée, une force et donner des effets. Le projectile utilise Map.move() pour se déplacer jusqu'à atteindre sa portée ou une cible puis elle disparaît.

Elément d'Objet. IV (Armures) :

- Creature.defense : sert à atténuer les dégâts adverses joueur possède une défense supérieure à 0.

Elément d'Objet.V (Amulette) :

- Hero.__init__() : Ajout d'un compteur de mouvement pour les amulettes
- Ajout de la classe Amulette.
- Amulette.regeneration() : Régénère la vie du héros tous les n mouvements.
- Amulette.coupCrit() : Tout les X tours on augmente significativement la force du héros pendant 1 tour afin d'infliger un coup critique.
- Game.play() : (Modification). A chaque mouvement on incrémente le compteur des amulettes afin d'appliquer les différents effets.

Salles

Elément de Salles. III (Boutique) :

- Création de la classe Marchand (qui hérite Element) qui permet d'acheter n'importe quel objet en échange d'or.
- Hero.argent : Argument d'instance qui permet de stocker l'or pour pouvoir acheter des objets chez le marchand.

Monstres

Elément de Monstres. III(Archers) :

Map.moveAllMonsters() : modification. Si la créature attaque à distance et que le héros est à portée alors elle tire dans la direction du héros au lieu de se déplacer.

Utilise la fonction handler.tir() qui est expliqué dans Elément d'Objet. III (Armes de Jet).

Modifications apportées par notre imagination :

Gameplay

Elément de Gameplay (sélection de héros) :

- Game.selectPerso() : Demande au joueur quel héros veut-il choisir. Tous ont des caractéristiques différentes qui peuvent être lu en appuyant sur "i".

Elément de Gameplay (score) :

- Game.play(), Map.move() : Modification afin d’afficher un score en fin de partie. Il se calcule avec le nombre d’étage atteint et le nombre de créatures vaincues.

Objet

Elément d’Objet. (Pierre philosophale) :

- Creature.vie : sert à pouvoir “ressusciter” la créature si sa vie atteint 0 en lui redonnant toute sa vie si la créature a plus d’une vie
- Handler.revive() : c’est la méthode qui permet de donner 1 vie à la créature

Elément d’Objet. (Effets (poison/freeze)) :

- Map.moveAllMonster() : Certaines armes permettent d’empoisonner ou d’empêcher l’ennemi de bouger. Cela se produit grâce à une probabilité et pendant un nombre de tour défini. L’effet de freeze et de poison sur les créatures se passe dans moveAllMonster. Le poison du héros se passe dans Game.play().

Elément d’Objet. (Full boost) :

- Handler.fullBoost() : c’est la méthode qui permet de d’augmenter de 5 les hp et hpMax de 2 l’attaque et la défense

Salles

Elément de Salles. (Forgeron) :

- Création de la classe Forgeron qui hérite de la classe élément qui permet d’améliorer une arme ou armure jamais améliorer en augmentant sa statistique de 1 en donnant 1 d’acier.
- Hero.acier : permet de stocker le nombre d’acier que l’on a pour ensuite pouvoir améliorer une arme ou armure.

Monstres

Elément de Monstres. (Boss) :

- Création de la classe Boss : Permet de différencier les monstre classique et les boss
- Boss.spawArmy() : Méthode qui permet au boss d’invoquer une créature aléatoire tous les n tour.
- Boss.tpBoss() : Permet au boss de se téléporter
- Map.move() : (Modification). Ajout d’une fonctionnalité. Si un certain boss subit des dégâts (ici Thanos), il se téléporte.
- Game.boss : liste qui contient les boss et leur type
- Game.buildBoss() : Ajoute plus ou moins de PV et de Force au boss en fonction de l’étage
- Game.buildFloor() : (Modification). On ajoute le boss au lieu de l’escalier (On doit tuer le boss pour accéder à l’escalier.
- Game.buildStairs() : (Modification). Ajoute un escalier au centre de la dernière salle ou sur des points à proximité.
- Map.play() : (Modification). A chaque fois que le héros se déplace n fois, un certain boss (ici Joker) invoque ses sbires.

Répartition du travail :

Timothé :

- Chef de projet.
- Optimisation de diverses lignes de codes
- Nuage de visibilité +
- Débugueur, compilateur.
- Menu pause, possibilité de sortir d'interface avec une touche d'exit,
- Optimisation de l'affectation des effets et du forgeron,
- Drop des monstres

Alexis :

- Effets de poison et de gel pour le Héro et les Créatures.
- Projectile.
- Objets utilisant des projectiles (arc, arc empoisonné, arc de gel).
- Créatures qui donnent des effets au héros (archer, archer empoisonné, mage de glace).
- Créatures qui tirent à distance (dragon).
- Score.
- Déplacement des créatures.
- Sélection des personnages.
- Optimisation du marchand.

Andy :

- Création des Amulettes et de leur capacité.
- Création des Armes
- Gestion des équipements (body, use, unEquip ainsi que les fonctions boost et nerf).
- Inventaire limité.
- Amélioration de la fonction heal
- Création des Boss, de leur implémentation ainsi que de leur capacité.

Titouan :

- Création des armures (armure d'or, d'argent et de bronze) utilisant la statistique défense.
- Forgeron améliorant une arme ou armure.
- Marchand vendant un objet.
- Quelques objets : FullBoost, pierre philosophale.