Last man in Vietnam

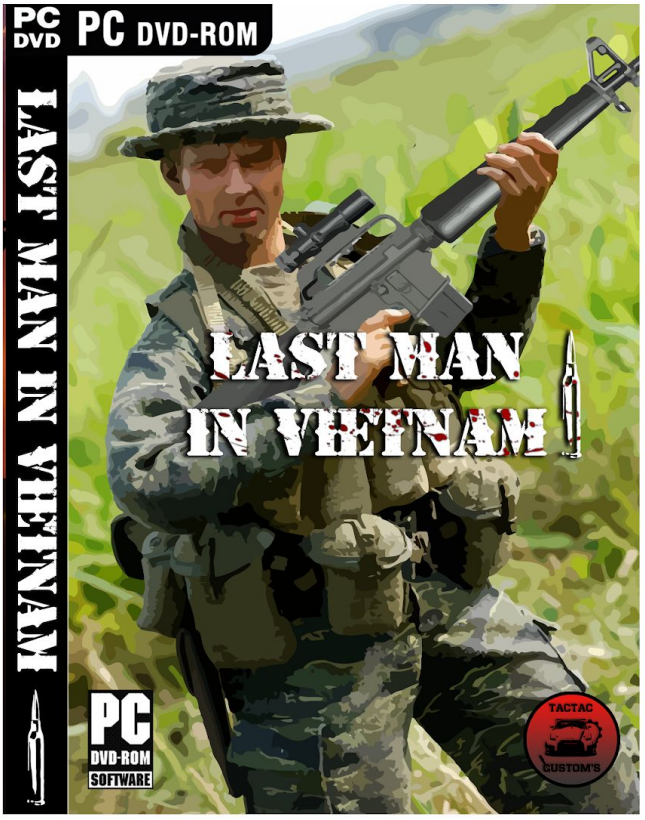
Par

Florian Pasquier - Romain Kamiri - Steven San

Juin 2019

Sommaire

Histoire du jeu

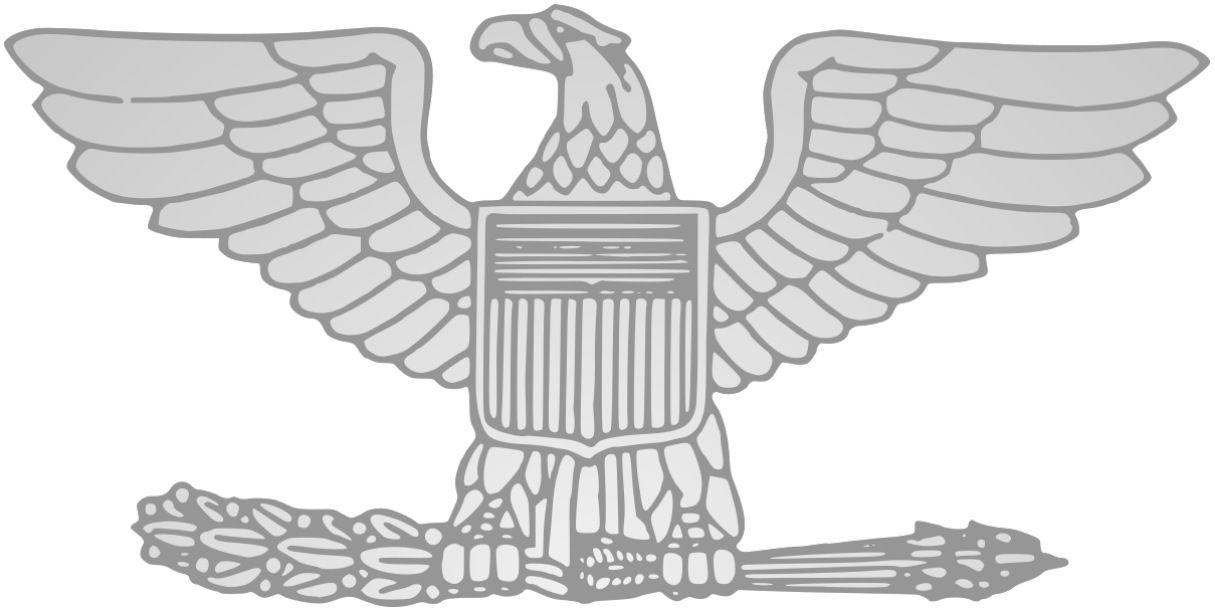


Bill Kilgore

Bill Kilgore est un ancien béret vert américain, le dernier survivant d'un commando d'élite formé durant la guerre du Viêt Nam et dirigé par le colonel Jack. Après la guerre du Viêt Nam où il a, entre autres, été capturé et torturé par l'ennemi, Bill Kilgore est de retour aux États-Unis. Il se heurte à l'hostilité de l'Amérique envers les anciens soldats et n'arrive pas à se réinsérer socialement.

Alors qu'il rend visite à son ancienne unité, il apprend la défection d’un membre de la CIA REETY Hughes john.

Colonel jack



le colonel jack est le deuxième commandant des bérets verts, il sert à bill de contact radio principal lui donnant des informations sur l'objectif de la mission et des conseils généraux sur le gameplay.

REETY Hughe John

Ancien membre de la CIA, envoyé au vietnam au début de la guerre, il a décidé de rejoindre les forces nord vietnamiennes, il n'est pas encore sur aujourd'hui de quelle fut la raison de sa trahison, mais il est suspecté qu'il ait été capturé et torturé afin de lui effectuer un lavage de cerveau et de le retourner contre les USA.

Le plastic gear mk1

Le Plastic Gear MK1 a été le premier tank entièrement bipède. Il a été conçu par les soviétique au vietnam en 1968 . Il est équipé d'un canon, d'un radar et de fusés pour accroître sa mobilité.

À l'origine une arme sans équipage, REETY Hughes John l'a détournée et l'a modifiée et prévoyait de lancer une bombe nucléaire sur la côte ouest des États-Unis si le président ne décidais pas de faire évacuer ses troupes du vietnam.

Le Vietnam

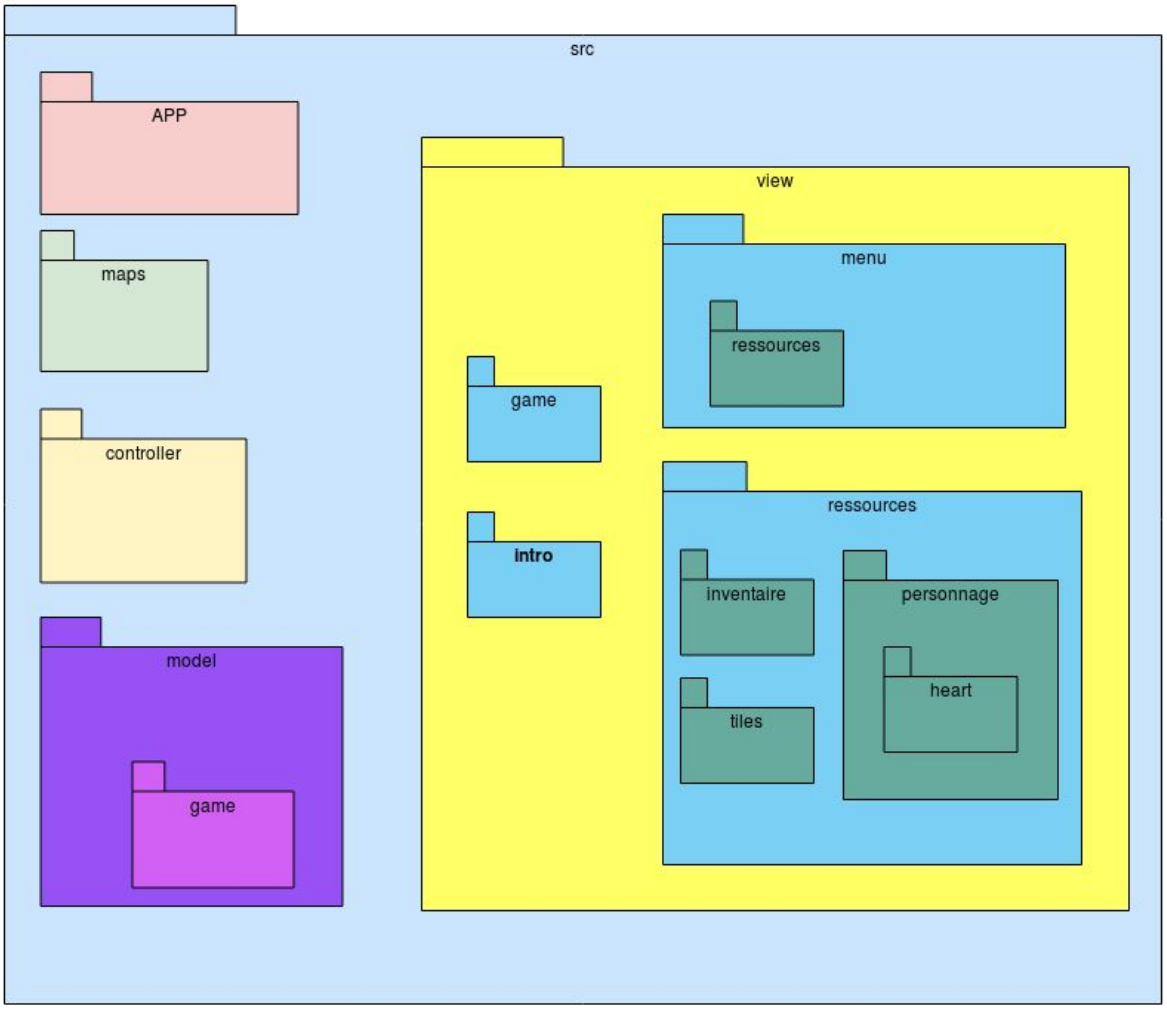
La guerre faisait rage au vietnam jusqu'à ce que REETY Hughes John ne menace nucléairement les États Unis, c'est pourquoi il n'y as plus de troupes américaines sur toute la partie nord du vietnam, et que le reste des troupe se préparent à rentrer au pays.
Bill Kilgore sera le seul soldat Américain à opérer dans cette partie du pays.

Le Contexte

Après avoir appris l'existence du plastic gear, le président des États-Unis décida de temporairement retirer ses troupes du vietnam pour éviter toute escalade nucléaire.
Hors ce dernier ne comptais pas se laisser faire et décida d'envoyer un seul homme pour détruire le plastic gear ainsi que son possesseur.
Bill est donc envoyé dans la région de //ici par hélicoptère et se doit d'accomplir sa mission en solitaire et sans équipement, tout ce qu'il aura sera récupéré sur place.

Documents pour la CPOO

Architecture

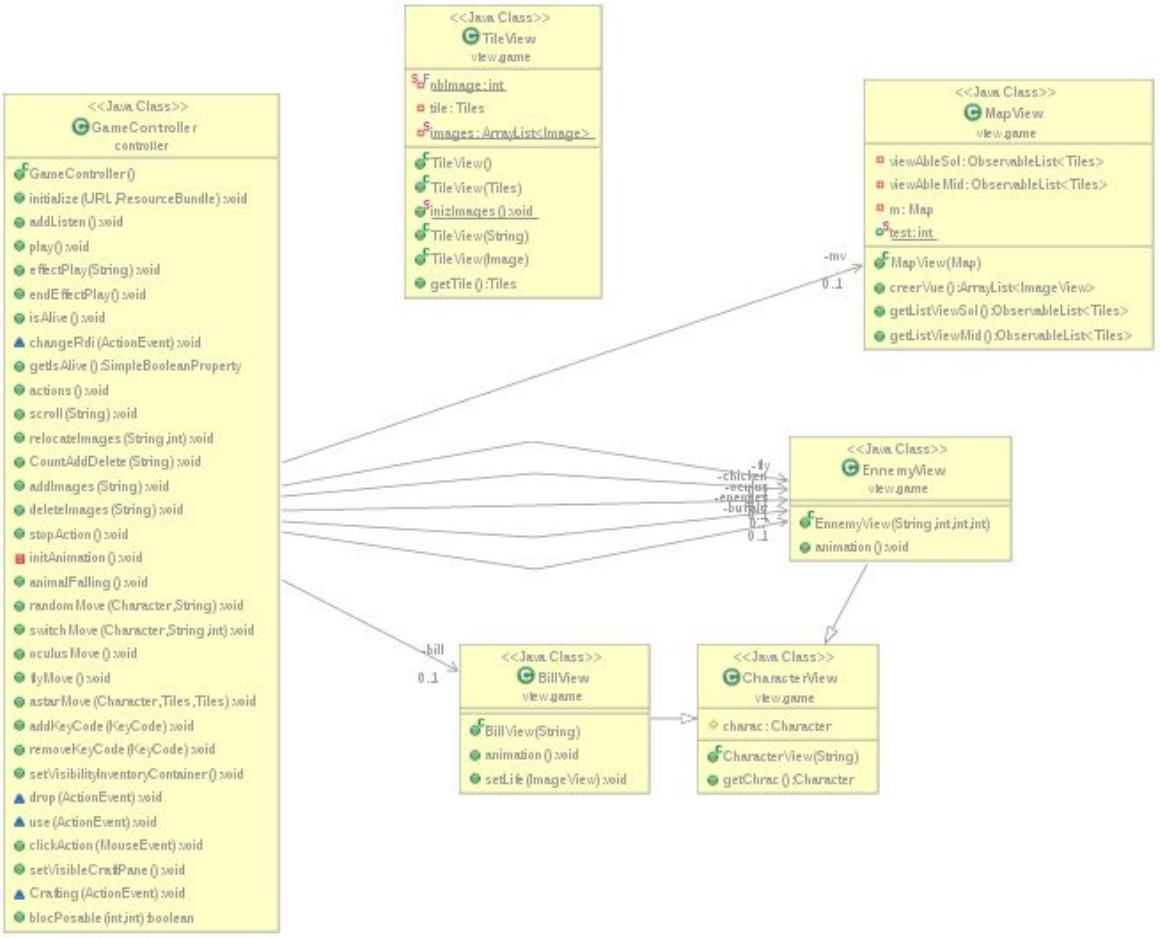


L'architecture global est divisé en 4 grands packages l'un gérant le main, l'un gérant les contrôleurs, un autre le modèle, un gérant la vue et enfin un petit package qui garde les fichier nécessaires à la map (JSON).

Le package de la vue est le plus imposant car il gère des classe "VUE" et des ressources à afficher (video, tiles, personnages …).

Le package controller contient les deux contrôleurs qui permettent de gérer le menu et le jeu.

Le package model contient l'ensemble des méthodes du jeu et est séparé des méthodes de la vue (afin de rester au maximum mvc).

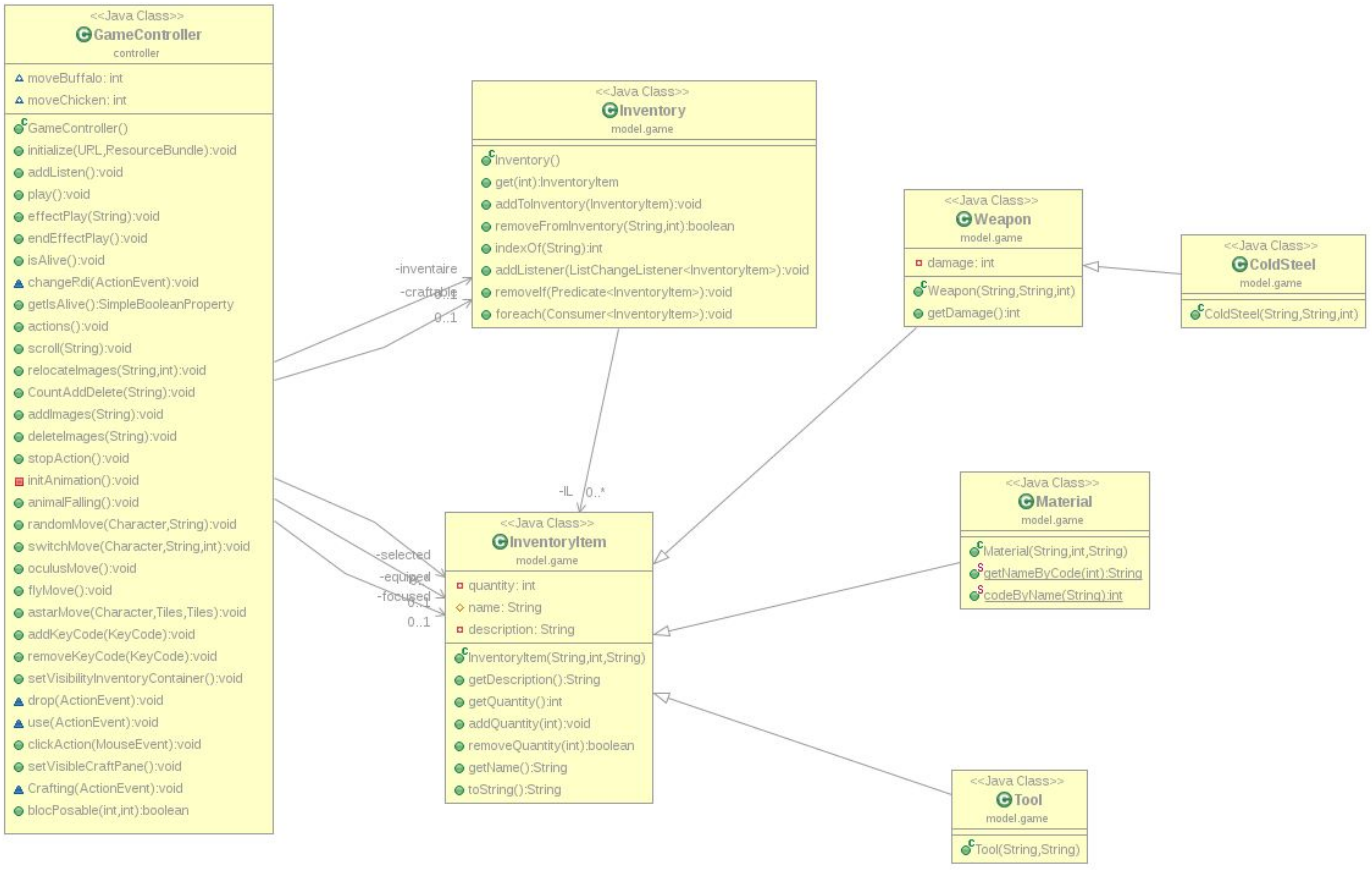


Exemple entre le contrôleur du jeu le package vue, on peut voir que la vue est utilisé par le controleurs afin d'afficher certaines images (tiles et personnages) sur la carte.

BillView et EnemyView sont des extensions de CharacterView, classe permettant de gérer chaque personnages du jeu en terme de vue, la classe TileView permet de gérer chacune des tiles du jeu et la class mapView sert à charger la carte d'afficher la partie visible de la carte.

Les déplacements sont gérés avec une liste de eventHandler afin de pouvoir se déplacer dans différentes direction (en sautant et se déplacent à gauche/droite) qui se trouve dans le contrôleur du jeu.

Détails : diagrammes de classe



Le diagramme représente les interactions entre le contrôleur, l'inventaire, et les items contenus dans l'inventaire.
Lorsqu'un item est ajouté (soit par le minage soit par le craft) l'inventaire (modèle) se met automatiquement à jour en créant un nouvel item (si aucun n'est contenu dans l'inventaire actuellement) soit en ajoutant la quantité récupérée à l'item en lui même.

Les items d'inventaires sont divisés en plusieurs catégories, les armes, les outils et les matériaux, eux même pouvant être divisés en plusieurs sous catégories (comme les armes blanches).

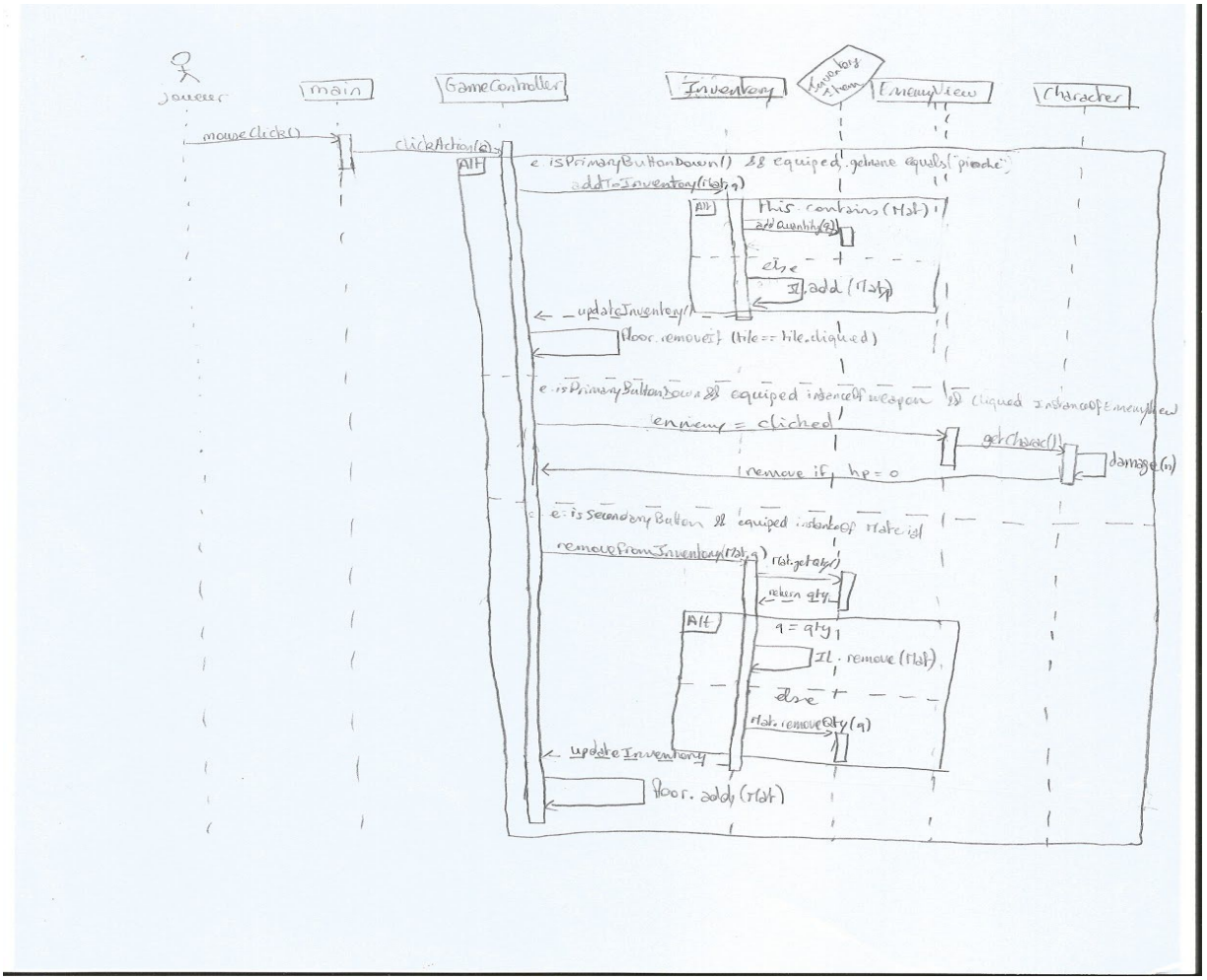
Des écouteurs permettent de mettre à jours la vue en fonction de ce qui est contenu dans l'inventaire automatiquement à chaque changement (contrôleur et le addListener() dans inventaire).

L'inventaire permet aussi d'afficher sur la vue les caractéristique des items contenus : leurs nom, description et quantité contenue dans l'inventaire.

Il est aussi possible de jeter un item de l'inventaire d'un coup dans le controlleur ce qui utilise la méthode removeFromInventory() afin de se délester d'un item inutile.

Diagrammes de séquence

Le diagramme de séquence détaille une action de souris émise par le joueur.
Lorsqu'un clic est détecté dans le jeu, la fonction clickAction() dans le GameController() est appelée.
Si le clic est un clic gauche et que le personnage est équipé d'une pioche, on ajoute à l'inventaire (contenu dans le modèle) le bloc miné et on supprime le bloc de la carte.
Si le clic est un clic gauche, que le personnage est équipé d'une arme et que l'on clique sur un ennemi, les points de vie de l'ennemi sont réduits. Si ses points de vie passent à 0, l'ennemis est retiré de la carte.
Si le clic est un clic droit et que le personnage est équipé d'un matériel, on enlève à l'inventaire le bloc posé et on ajoute le bloc à la carte.



Structures de données

Tableau(Map)

La carte est sous forme de tableau car il était beaucoup plus simple de gérer les collisions avec des indices clairs plutôt que par des calculs complexes, on y gagne aussi en vitesse de calcul.

Liste Observable(GameControlleur)

permet de gérer les tiles, les item d'inventaire et la vue des ennemies dans le contrôleur ainsi que leur affichage

HashMap(craft)
Le craft suit le principe d'une HashMap, en effet lorsque l'on lui donne une clé, il nous renvoie l'item correspondant.

Stack(A*)
Le stack a été utilisé pour gérer la file d'attente dans l'algorithme de déplacement A* car c'est une structure de donnée qui emploie la méthode LIFO (Last In First Out).
Emplacement des méthodes dans la classe collision.

ArrayList(radio, touches)
L'arrayList a été choisi pour gérer les communications radio, en gérant les lignes de texte à afficher car elle était une solution simple et adapté à un simple parcours.
Elles a aussi été choisi pour avoir une saisie de touches simultané (afin de pouvoir effectuer plusieurs action en même temps).

Exception

Chargement de la carte avec un try catch (MAP)
Le try catch permet de vérifier que la lecture du fichier (json) se passe correctement et dans le cas contraire de nous indiquer le problème survenue.

Exception dans le main (MAIN)
Le try catch est la pour vérifier que l'initialisation du FX fonctionne correctement, sinon de nous notifier d'un message d'erreur la source du problème.

Algo

A* (CollisionManager et GameControlleur)

Le A* permet trouver le chemin le plus court reliant les ennemis au personnage,
il est limité par le type d'ennemi dépendamment qu'il soit volant ou au sol, l'hélicoptère n'est pas limité par les sauts par exemple, tandis que le slime (au sol) ne peut pas sauter de plus d'un bloc de hauteur.
Il est aussi limité par une distance de son point de départ à partir duquel il suis/ ne suis plus le joueur.
Le A* à été choisit car il répondait au attentes d'un déplacement efficace sur la carte pour venir combattre le joueur, qu'il était peu demandeur en ressources et qu'il permettait d'être adapté à tous types d'ennemis.

Scrolling (Controller)

Le scrolling permet de faire apparaître les tiles en fonction des coordonnés du joueur, il permet de nettoyer la mémoire en ne gardant que ce qu'il doit être affiché sur l'écran et de ne pas avoir de problèmes de ralentissement dû à un trop grand nombre d'images affichés en hors champs.
Il charge en fonction de la direction et de la position du joueurs (virtuelle), tout en laissant le personnage du joueur au centre de l'écran, (il n'y a que la carte qui se déplace).

Algo de réarrangement de l'inventaire (GameController)

Cet algorithme est primordial pour l'inventaire, en effet il permet de réarranger la vue de ce dernier qui est en réalité un gridPane car il ne le fais pas automatiquement.
Le principe est de vérifier l'inventaire côté modèle (liste observable) et de redéfinir les contraintes des cases en fonction des colonnes et lignes concernés par le changement.

Fonctionnalités :

Pour se déplacer

Il suffit d'utiliser les touches Q, D, ou les flèches directionnels gauche et droite pour se déplacer à gauche ou à droite,ainsi que d'appuyer sur la touche ESPACE pour sauter.

Miner un bloc

Afin de miner un bloc, il faut s'équiper de la pioche à disposition et appuyer sur un bloc à proximité du personnage avec le clic gauche. Cela aura pour effet de faire disparaître le bloc ciblé et de le rajouter dans l'inventaire.

On ne peut miner que les blocs non traversables, c'est à dire les blocs situés sous les arbres.

Si le bloc se situe sous le sol, miner un bloc créera une grotte et le bloc miné se transforme en un autre bloc traversable qui serait le fond de la grotte.

Poser un bloc

Pour poser un bloc, il suffit de d'équiper un bloc dans l'inventaire puis de cliquer sur le terrain avec le clic droit et doit être à la portée du personnage. le bloc posé doit impérativement être fixé à un autre bloc (à droite, à gauche, en haut ou en bas), on ne peut pas poser de bloc dans le vide mais un bloc peut être en lévitation si on casse les blocs qui l'entourent.

Inventaire

La pioche est l'objet équipé par défaut.

L'inventaire s'ouvre avec la touche I, il contient tous les objets actuellement possédés par le personnage. En appuyant sur un objet, on peut le sélectionner. Un descriptif apparaîtra

ensuite à l'écran détaillant les spécificité de l'objet. Deux choix sont alors présent, équiper pour équiper l'objet ou jeter pour jeter toute la pile de l'objet sélectionné.

Crafting

pour ouvrir le menu de crafting, il faut appuyer sur la touche C. Sélectionner l'item à crafter

et d'appuyer sur le bouton CRAFTER. S'il y a pas assez de matériaux pour crafter l'item, l'item ne sera pas crafté.

On peut actuellement crafter un couteau avec 1 bois et 2 métal et un crayon avec 1 bois et 1 métal.

Attaquer un ennemi

Pour attaquer un ennemi, il faut d'abord être équipé d'une arme blanche que l'on aura crafter au préalable (Couteau ou Crayon). Il faut ensuite cliquer sur l'ennemi avec le clic gauche pour faire baisser ses points de vie, plusieurs coups seront nécessaire pour le tuer et le faire disparaître. A noter qu'il faut aussi être à proximité de l'ennemi pour l'attaquer.