Dossier du Projet

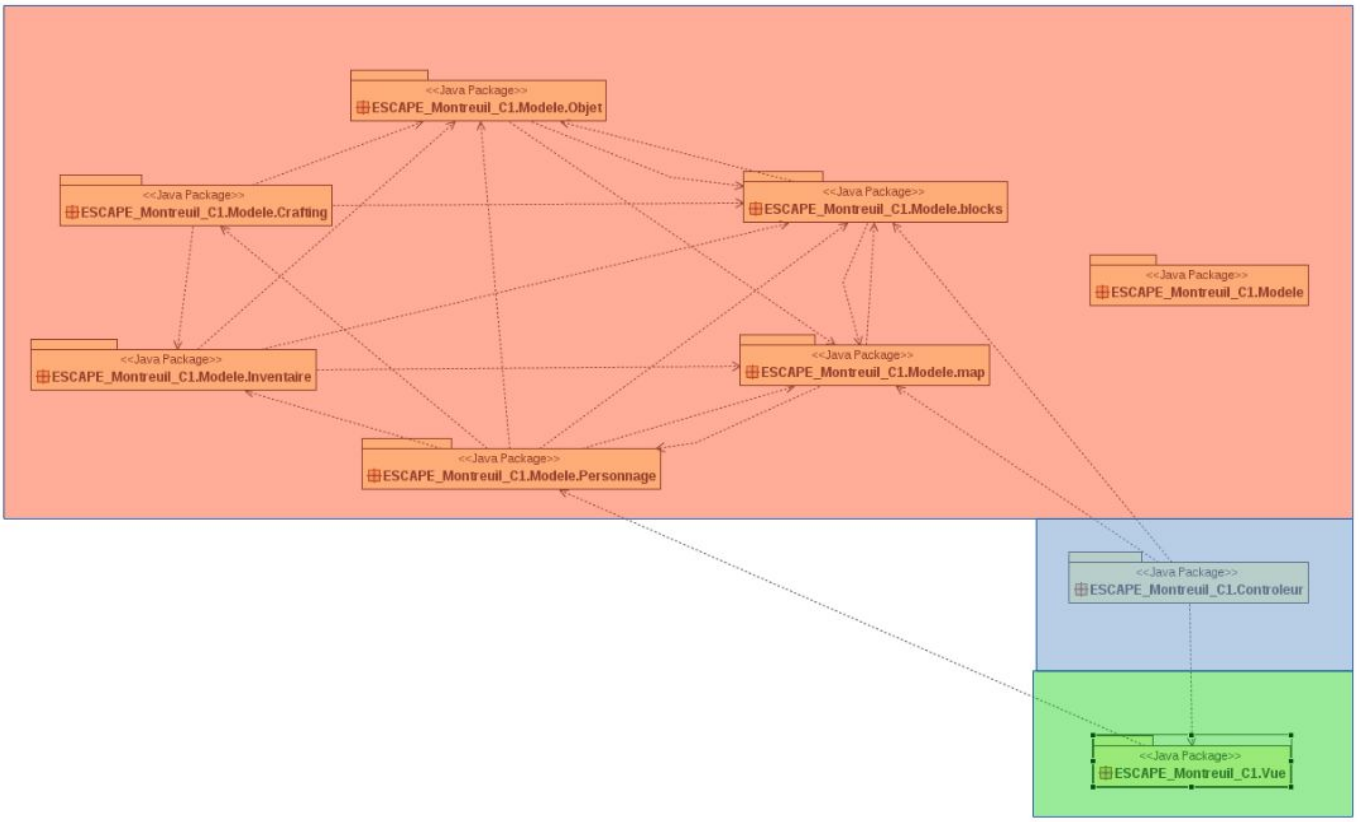
GROUPE : Rabie Atta, Yingwang Zhou, FILS-AIME Owenn

1:Documents pour CPOO

Architecture (5/60 points)

Diagramme de Paquetage

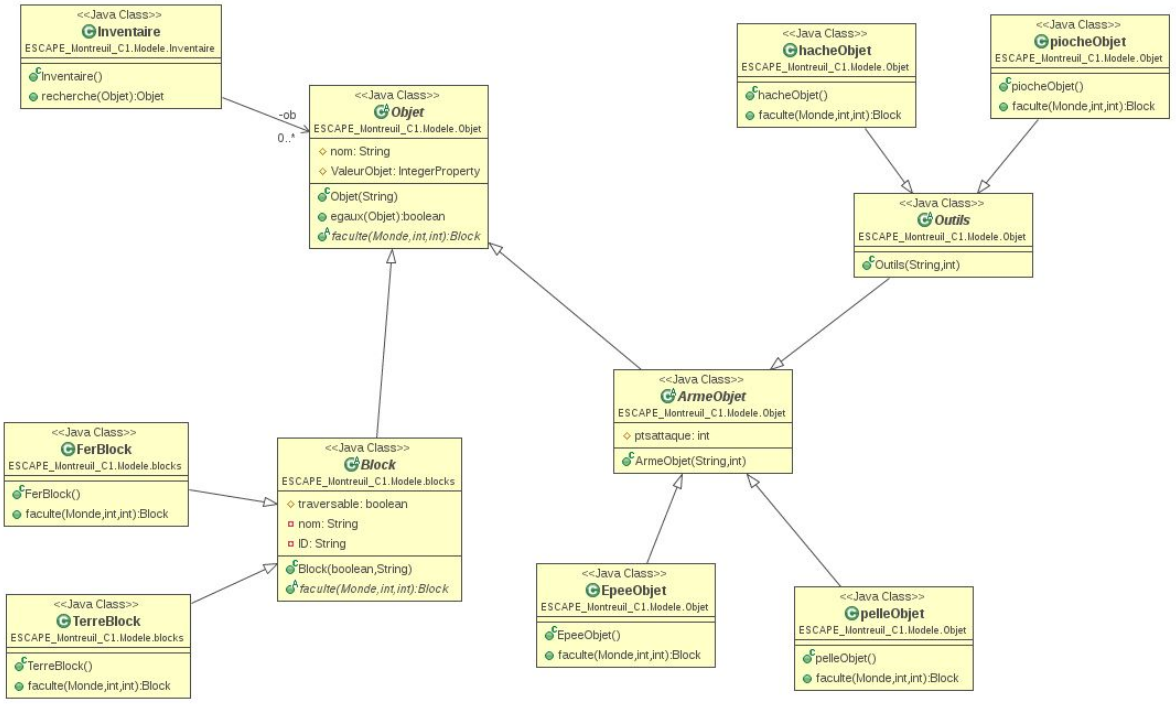
Rouge = Modele
Bleu = Controlleur
Vert = Vue



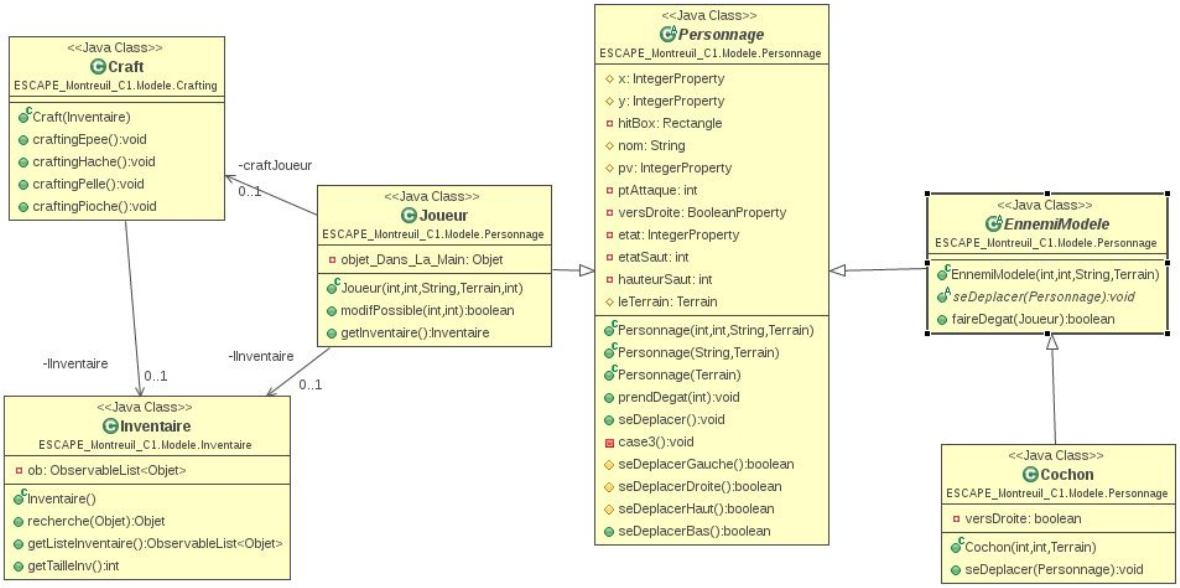
Nous pouvons remarquer que le contrôleur fait le lien entre le modèle et la vue. En effet le modèle fait tourner le programme, la vue permet d'afficher le programme et de prendre en compte les interactions avec le joueur. Enfin le contrôleur est l'intermédiaire entre ces deux

derniers par exemple il permet de définir les images selon les blocks, exécuter des fonctions selon les choix du joueur ou encore gérer la gameloop.

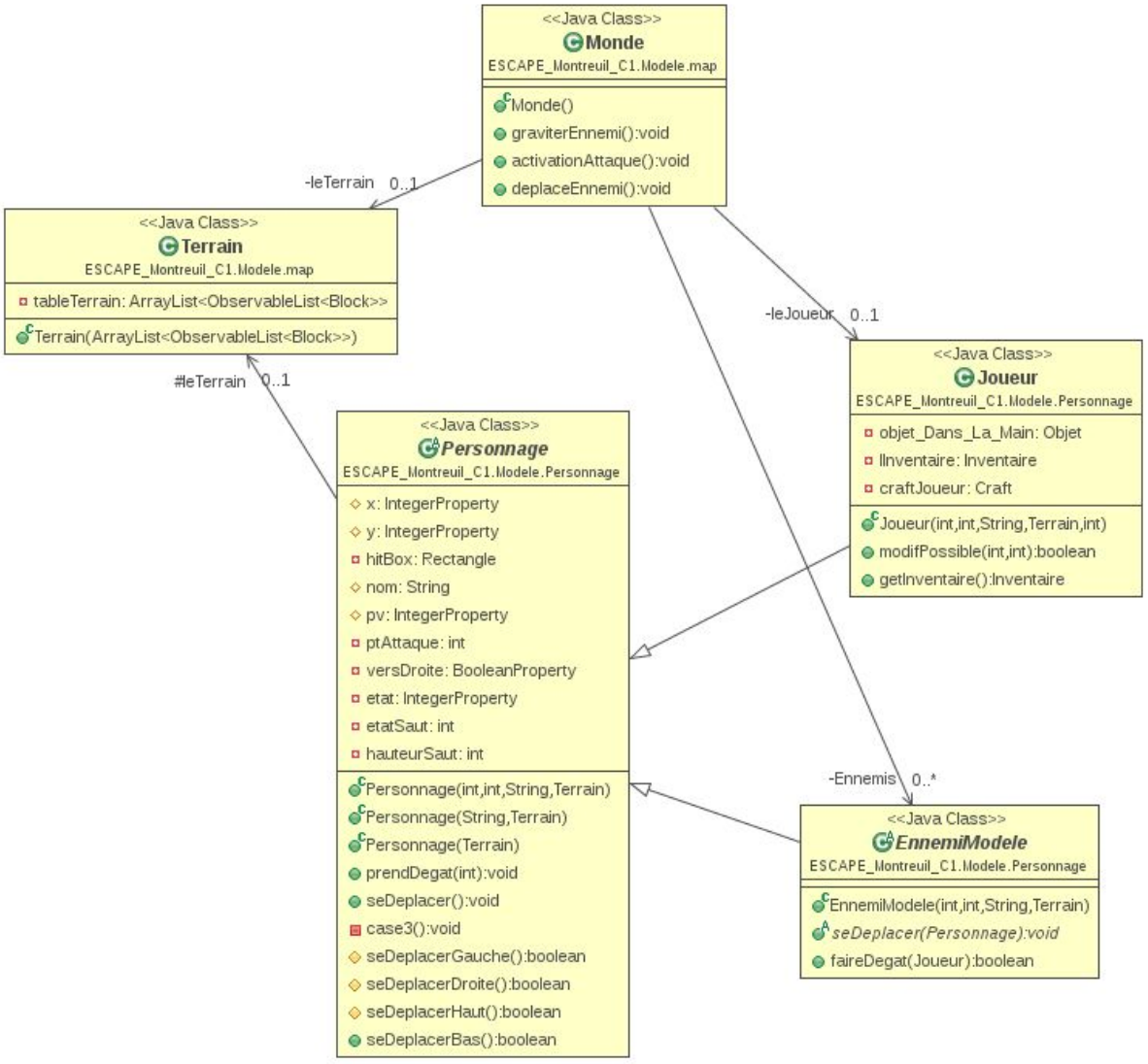
<u>Détails : diagrammes de classe</u> (5/60points)



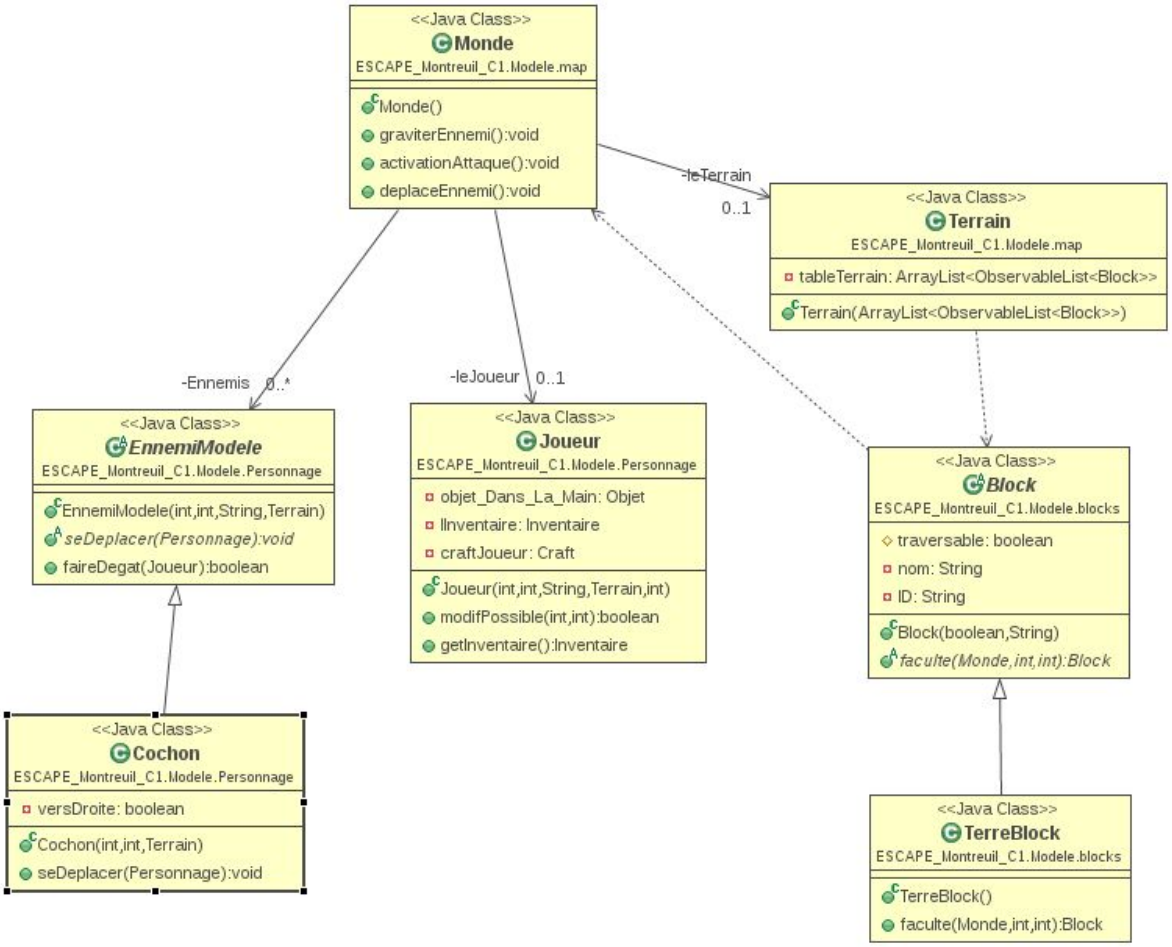
Ce diagramme représente le fonctionnement de l'inventaire. En effet, l'inventaire est composé d'Objet, des "Block" pour les ressources et des "ArmeObjet" pour les autres éléments de l'inventaire.
Chaque Objet a une faculté, en ce qui concerne les ressources leur faculté est de pouvoir être posé et pour les "ArmeObjet" est d'attaquer ou creuser.



Le diagramme ci-dessus, montre comment nous avons pensé notre code concernant le personnage. Nous avons choisi cette hiérarchie car elle permet au joueur de crafter et les Ennemis de se déplacer de manière indépendante toute en évitant une redondance évidente de code. Le craft possède l'inventaire du joueur dans lequel il peut effectuer des modifications sur les différentes quantités (valeurs). Le "Joueur" et les "Ennemis" sont des Personnages.



Grâce à cette hiérarchie, le personnage peut se déplacer en prenant en compte les positions des blocs du Terrain. Cela est possible car le personnage “connaît”(possède) le terrain. Nous pouvons également voir que le monde possède le joueur et une liste d’ennemi ce qui permet une manipulation plus aisé des ennemis.



Sur ce diagramme, nous pouvons voir la classe Monde qui essentiel au bon déroulement du Jeu, est qui permet d'éviter d'avoir dans le contrôleur le personnage, les ennemis… Et ainsi éviter de se tromper. De plus sémantiquement, il est logique de regrouper ces informations dans une seule classe "Monde".

Diagrammes de séquence (5/60 points)

Joint à ce document. Vous trouverez le diagramme de séquence de la destruction de bloc.

Structures de données :

Nous utilisons une HashMap dans le contrôleur qui nous permet de récupérer les images de manière plus simple.

```
public void creerDictionnaire() {
    dictionnaireImage = new HashMap< String,Image>();
    dictionnaireImage.put("A",new Image("ESCAPE_Montreuil_C1/source/air.jpg" ));
    dictionnaireImage.put("t", new Image("ESCAPE_Montreuil_C1/source/terre.jpg"));
    dictionnaireImage.put("B", new Image("ESCAPE_Montreuil_C1/source/tronc.jpg"));
    dictionnaireImage.put("I", new Image("ESCAPE_Montreuil_C1/source/Fer.png"));
    dictionnaireImage.put("T",new Image("ESCAPE_Montreuil_C1/source/solpetit.jpg"));
    dictionnaireImage.put("f",new Image("ESCAPE_Montreuil_C1/source/feuille.jpg"));
    dictionnaireImage.put("F",new Image("ESCAPE_Montreuil_C1/source/tronc.jpg"));
    dictionnaireImage.put("P",new Image("ESCAPE_Montreuil_C1/source/pierre.jpg"));
    dictionnaireImage.put("p",new Image("ESCAPE_Montreuil_C1/source/pioche.jpg"));
    dictionnaireImage.put("H",new Image("ESCAPE_Montreuil_C1/source/hache.jpg"));
    dictionnaireImage.put("e",new Image("ESCAPE_Montreuil_C1/source/épée.jpg"));
    dictionnaireImage.put("L",new Image("ESCAPE_Montreuil_C1/source/pelle.jpg"));
}
```

Exception :

Nous utilisons des exceptions dans la lecture du fichier CSV en effet dans la lecture de buffer nous utilisons des try catch qui nous permet d'éviter tels que erreur liée au fichier comme introuvable, null pointer exception.

```
/**
 * Le MapReader permet de traduire la map csv en liste de list observable de Block
 */
//constructeur
public MapReader() {
    try {
        this.fp = new File("src/ESCAPE_Montreuil_C1/source/map5.csv");

        this.fpr = new FileReader(this.fp);
        this.bfReader = new BufferedReader(this.fpr);
    }catch(FileNotFoundException e1) {
        System.out.println("Erreur: le fichier est introuvable");
    }
    catch(IOException e) {
        System.out.println("Erreur le fichier ne peut etre ouvert");
    }
    this.terrain=new ArrayList<ObservableList <Block>>();
}

//methode
public void constructeurMap() {
    int i=0;
    String [] ligne;
    try {
        //un faire tant que on est pas arriver a la fin
        do {
            ligne=this.bfReader.readLine().split(","); //separe le resultat en tableaux de String en fonction des point virgule
            if(ligne!=null) {
                lireLigne(i,ligne);
            }
            i++;
        }while(ligne!=null);
    }catch(IOException e) {
        System.out.println("Erreur: le fichier introuvable");
    }catch(NullPointerException e) {
        System.out.println("Fin de la lecture");
    }
}
```

Algo :

Il y a un Algo pour les déplacements des cochon et des oiseaux dans le package "ESCAPE_Montreuil_C1.Modele.Personnage"

Le cochon va vers la droite lorsqu'il ne peut plus aller vers la droite il va vers la gauche. De plus il saute de manière aléatoire.
L'oiseau bouge de gauche à droite comme le cochon et attaque le Joueur seulement lorsque le Joueur entre dans la zone "unsafe"(la montagne qui sépare les cochon des lamas), a partir de ce moment-là, l'oiseaux suit le Joueur jusqu'à ce que celui-ci quitte la zone "unsafe".
Les Ennemis attaquent le joueur lorsque celui-ci se trouve à proximité.

2 :Documents pour gestion de projet

Document utilisateur (10 points/20)

Concept et univers

Le personnage est le seul survivant d'un crash d'avion il se retrouve dans une immense forêt. Dans cette forêt le personnage se trouvera face à différents obstacles et ennemis. Il a pour objectif de survivre, récolter des ressources, fabriquer des objets pour réparer l'avion et repartir.

Les fonctionnalités et mécaniques du jeu

- Déplacements : Pour déplacer le joueur utiliser "Q Z D" ou les flèches du claviers. Z et la flèche du haut pour sauter, la touche Q et la touche gauche pour aller vers la gauche. Et la touche D et la touche droite pour aller à droite.

- Crafter : Pour crafter vous devez d'abord ouvrir la table de crafting à l'aide de la touche "i", puis appuyer sur le bouton créer, mais avant cela il faut vérifier que vous avez suffisamment de ressource. L'objet crafter sera directement ajouter dans l'inventaire mais on peut en avoir qu'un dans l'inventaire.

Voici les ressources nécessaires :

Tableau de craft

	Bois	Pierre	Fer
Pelle	0	0	0
Hache	2	3	0

Pioche	2	0	3
Épée	3	0	5

- Miner et Poser : Pour miner un bloc, il faut choisir l'objet qui le permet dans l'inventaire à l'aide des touches allant de F5 à F8 puis clique gauche sur le bloc à récolter. Pour déposer une ressource, il faut commencer par sélectionner la ressource en question. A l'aide des touches allant de F1 à F4 cliquer sur le lieu souhaité. Peu importe l'action que vous voulez faire, elle ne peut s'effectuer que dans une distance d'un bloc du joueur.

Tableau de d'utilisation

Voici un tableau présentant les Objet et l'élément qui peuvent casser

Nom casse block	Bois	Pierre	Fer	Terre
Pioche = F5		x	x	
Hache = F6	x		x	
Pelle = F8	x	x		x