



HORDE

Nicolas Carbonnier | Anne Laprevote | Tom Wysocki
19 Decembre 2018



PRESENTATION DES CLASSES

HORDE

[Met en place la partie : génère la carte, crée les citoyens. Possède la boucle de jeu et gère toutes les interactions avec l'utilisateur]

Méthodes

public void init() : génère la carte, ajoute les items (Planches, Métal, Boissons Energisantes) sur la Carte, crée les deux **ArrayList** qui correspondent aux Actions Ville/Exterieur.

public void jouer() : boucle de jeu (demande les actions à tour de rôle aux joueurs en fonction de leur position (Ville/Exterieur))

private boolean actionVille(Citoyen _citoyen, int _action)

private boolean actionExterieur(Citoyen _citoyen, int _action): effectue l'action demandée par le joueur en fonction de la valeur saisie. Retourne VRAI sauf si l'action demandée est 'passer son tour' (0).



Attributs

Case [][]carte : Carte du jeu

Talkie talkie : talkie walkie

int nbCitoyens : nombre de joueurs

ArrayList<Citoyen> citoyens : liste qui contient tous les joueurs

ArrayList<String> actionVille : liste des actions pour la Ville

ArrayList<String> actionExterieur : liste des actions pour l'Exterieur



« Houla, je ne mourrai pas d'une crise cardiaque moi » [Horde]

CITOYEN

[Représente un citoyen, permet d'effectuer toutes les actions associées au citoyen]

Attributs

`Case[][] carte` : Carte du jeu
`Talkie talkie` : talkie walkie
`String nom` : noms des joueurs
`int pv, pa` : point(s) de vie et point(s) d'actions du joueur
`int x, y` : coordonnées du joueur sur la carte
`boolean enVille` : VRAI si le joueur est en ville FAUX sinon
`int nbPlanches, nbMetal, nbBoissons, nbGourde, nbRation` : nombre par item dans l'inventaire (le total ne peut pas dépasser 10)
`boolean aBu, aMange` : passe à VRAI lorsqu'un joueur utilise une Ration ou une Gourde(duit), repasse à FAUX à la fin de journée
`boolean accroc` : passe à VRAI une fois qu'un joueur boit une Boisson énergisante
`int dose` : compteur de tours avant de perdre des `pv` (Boissons énergisantes)

Méthodes

`public void deplacer(int _direction)` : déplace un joueur dans une direction (conditions: le joueur a au moins un `pa`, et n'est pas en extrémité de carte) et fait appel en fonction de la direction choisie à :

- `deplacerN()` : NORD
- `deplacerS()` : SUD
- `deplacerE()` : EST
- `deplacerO()` : OUEST

`public void mangerRation() et boireGourde()` : permettent de manger une Ration ou boire une Gourde si le joueur en possède une et s'il n'en a pas déjà consommé une dans la journée.

`public void finDeTour()` : redonne 4 `pa` au Joueur (sans dépasser 10 au total), gère l'addition aux Boissons énergisantes

`public void finDeJournée()` : repasse `aBu` et `aManger` à FAUX

EN VILLE

`public void ouvrirPorte() et fermerPorte()` : permet d'ouvrir ou de fermer la porte de la ville si ce n'est pas déjà fait

`public void prendreRation() et remplirGourde()` : permet de stocker en inventaire UNE ration (prise dans l'entrepot) et UNE gourde(prise au puit)

`public void allerAuPuit()` : récupère directement 6 `pa` (UNE fois par jour)

`public void déposerPlanches() et déposerBoisson(int_qt)` : permet de déposer dans l'entrepot les items choisis. Dépose l'entièreté des Planches et du Métal de l'inventaire, sélection de la quantité pour les Boissons.

`public void construire(int _id, int _pa)` : fait progresser la construction de la défense `_id` (index de la liste des défenses [Classe Ville]) avec `_pa`

`public void nouvelleDefense(int _id)` : crée une nouvelle défense (`_id`: index dans l'énumération Défenses [Classe Defense])

EXTERIEUR

`public void fouiller()` : fouille la case (si assez de `pa`) et rend visible son contenu à tout joueur présent sur la case

`public void prendrePlanches, prendreMetal, prendreBoissons(int _qt)` : récupère le nombre saisi d'item sélectionné (si le nombre est inférieur ou égal au nombre restant sur la case et si la place dans l'inventaire est suffisante)

`public void attaquerZombie()` : attaque un zombie s'il en reste sur la case et si le joueur a assez de `pa` avec une probabilité de 10% de perdre 10 points de vie (pv).

`public void majTalkie()` : utilise un `pa`, met à jour la case (ou la rend visible) du joueur sur le Talkie Walkie (Boissons énergisantes)

« - Tu vas risquer nos vies pour un Twinkie ?

- Y'a une boîte de Twinkie dans ce supermarché. Pas n'importe quelle boîte de Twinkie. La dernière boîte de Twinkie que quiconque pourra savourer dans tout l'univers. Et crois le ou non mais les Twinkie ont une date de péremption ok ? Dans un avenir très rapproché, la vie de ce délicieux rondin de sucre sera terminée. Ok, alors maintenant courage ou dégage le p'tit »

[Bienvenue à Zombieland, Columbus et Tallahassee]



« Venez vieilles peaux » [Horde]

CASE

[Définit une case selon ses coordonnées dans le tableau à double dimension]

Atributs

`int x,y` : coordonnées de la case



EXTERIEUR_[étendue de Case]
[Définit une case extérieure, permet les actions sur la Case]

Attributs

int nbPlanches, nbMetal, nbBoissons : nombre d'item sur la case
int nbZombies: nombre de zombies [""vivants""] sur la case
boolean fouillee : passe à VRAI une fois que la case est fouillée

« How do you kill someone that is already dead ? » [Braindead]

Méthodes

public void ajouterPlanche(), ajouterMetal(), ajouterBoisson() : utilisée lors de la génération de la carte uniquement, ajoute un par un un item sur la case (répartition random)
public boolean fouiller() : permet de fouiller une case
public boolean prendrePlanches, prendreMetal, prendreBoissons(**int** _qt) : permet de prendre la quantité sélectionnée par le joueur et le met dans son inventaire (si quantité sélectionnée valide)
public boolean attaquerZombie() : tue un zombie sur la case
public boolean encoreZombie() : retourne VRAI s'il reste un zombie sur la case

VILLE_[étendue de Case]
[Définit la Ville avec son Entrepôt et ses Défenses, permet les actions en Ville]

Attributs

boolean porteOuverte : VRAI si ouverte
int nbPlanches, nbMetal, nbBoissons, nbRations : nombre d'item correspondant dans l'entrepot
ArrayList<Defense> defenses : liste des défenses de la ville

Méthodes

public boolean ouvrirPorte(), fermerPorte() : permet d'ouvrir/fermer la porte
public boolean prendreRation() : prend UNE ration dans l'entrepôt (retire de l'entrepôt)
public void déposerPlanches, déposerMetal, déposerBoissons(**int** _qt) : dépose _qt item dans l'entrepot
public boolean construire(**int** _id, **int** _pa) : fait progresser la construction de la défense _id (index dans la liste des defenses) avec _pa
public boolean nouvelleDefense(**int** _id) : crée et ajoute une nouvelle defense à la liste des défenses de la Ville s'il y a suffisamment de ressources dans l'entrepot (_id: index dans l'énumération Defenses[Classe Defense])
public int defenses() : retourne la quantité de zombie auxquelles la ville peut résister



«Ho ! Ça ? C'est l'état actuel de tes défenses... alors bouge toi le c** et construis ! [joueur1]

DEFENSE

[Définit une défense, contient l'énumération des défenses possibles]

Attributs

String nom : nom de la défense
int nbPlanche, nbMetal : nombre de ressource nécessaires
int nbPA : nombre de **pa** restants avant que la défense soit active
int nbZombie : nombre de zombie auxquels la défense peut résister
boolean active : passe à VRAI une fois que **nbPa** atteint 0
static enum Defenses : énumération des défenses possibles (exemple: MUR(«Mur d'enceinte», 20, 5, 10, 20))

Méthodes

public boolean construire(int _pa) : retire **_pa** au nombre de **pa** restants avant que la défense soit active (sauf si **_pa** > **nbPa**), fait passer **active** à VRAI lorsque **nbPa** atteint 0
public static String listeDefenses() : retourne les défenses de l'énumération sous forme d'une chaîne de caractère

TALKIE

[Correspond au Talkie-Walkie, permet de mettre à jour les cases sur la carte visible]

Attributs

String [][] carteInfo :
 - case non fouillée & non mise à jour : «#####»
 - case fouillée & mise à jour : chaîne correspondant au nombre d'item de la case «
Case [][] carte : carte du jeu

Méthodes

public boolean miseAJour(int _x, int _y) : modifie la chaîne de **carteInfo[_y][_x]** en fonction des items contenus de la case (**carte[_y][_x]**) si la case a été fouillée



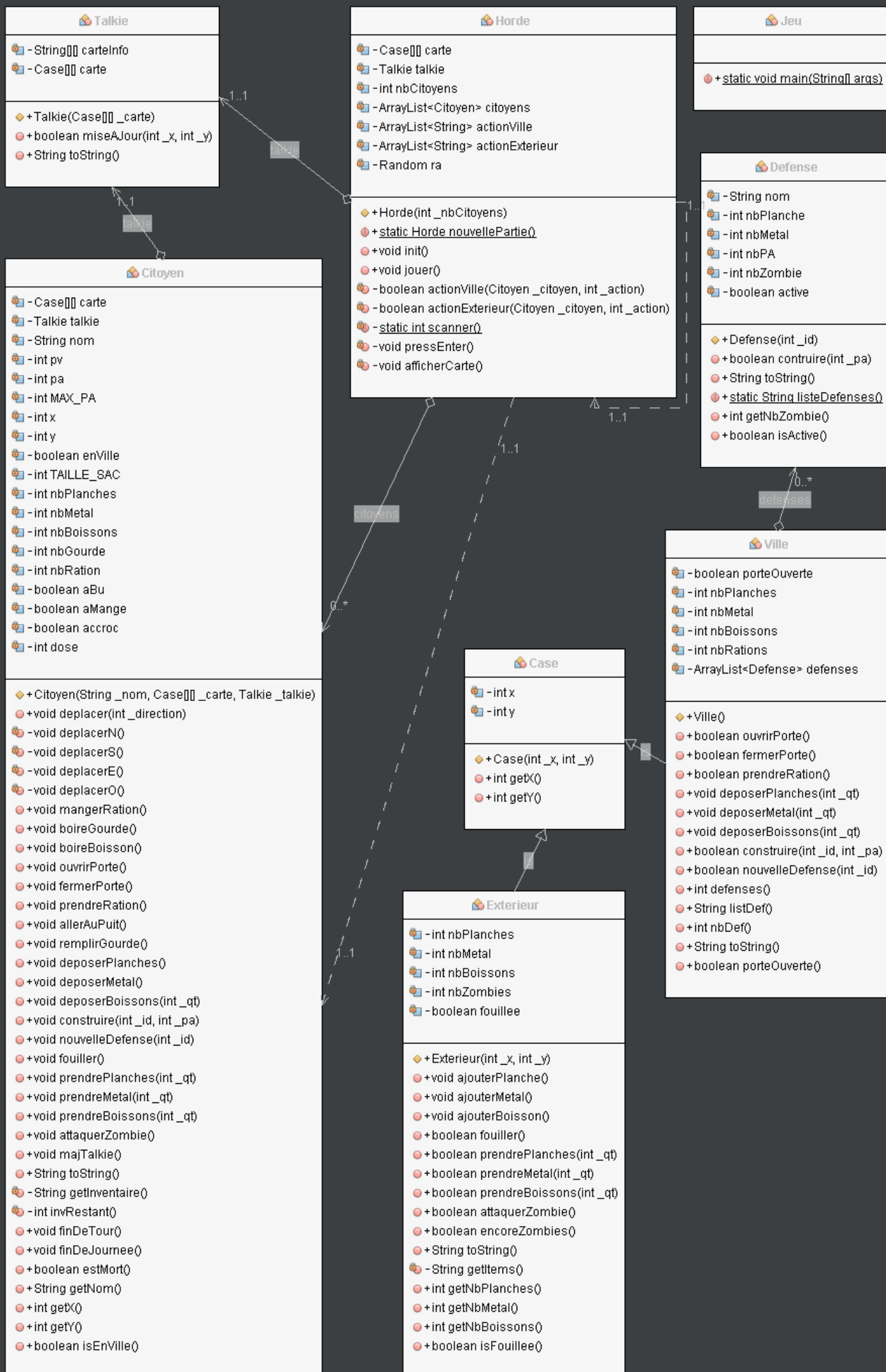
DÉMARRAGE DU JEU

READY TO DIE?

Demarrage de la classe Jeu (Main) [...]
Invitation du nombre de suicidaires à jouer...
Generation de la carte [...]
Demande les noms des joueurs **FOR THE GRAVE**
Démarrage de la boucle de Jeu [...]

Ready ?
FAREWELL!

**COPS YOU'VE BEEN HACKED AND
YOU'LL STAY THERE FOR EVER!**





BONUS



Meilleures citations de zombies :

«Bwwaaaaaaaaarrrrr»

«Gnnnnnnnnnnnn»

«Yerrrrrrrr»

«Grrrrrrrrrrrr»

«Rrrrrrrrrrr»

Et pour les plus développés : «Braiiiiiiiiins»

«- Écoute j'suis pas... j'suis pas très doué pour les adieux alors... « Ça c'est un bon cochon »...
- C'est vraiment le pire au revoir que j'ai entendu. Et en plus c'est une réplique de film. »

[Bienvenue à Zombieland, Tallahassee à Columbus]

