

# GestionCombat

Kevin Coquart, Quentin Bunel

18 avril 2014

## Contents

### 1 GestionCombat

#### 1.1 service : GestionCombat

#### 1.2 use : Personnage, Position

#### 1.3 types : String, int, boolean, List<Personnage>, Map<String,Personnage>, Map<String,COMMANDE>, enum COMMANDE {RIEN, GAUCHE, DROITE, HAUT, BAS, SAUTER, FRAPPE, JETER, RAMASSER}

#### 1.4 observators :

##### 1.4.1

- terrain : [GestionCombat]  $\rightarrow$  Terrain
- mPerso : [GestionCombat]  $\rightarrow$  Map<String,Personnage>
- estFrappe : [GestionCombat]  $\times$  String  $\rightarrow$  boolean
  - pre estFrappe(C,id) require id  $\in$  mPerso.keySet()

- $\text{estGele} : [\text{GestionCombat}] \times \text{String} \rightarrow \text{boolean}$ 
  - $\text{pre } \text{estGele}(C, \text{id}) \text{ require } \text{id} \in \text{mPerso.keySet}()$
- $\text{cptGele} : [\text{GestionCombat}] \times \text{String} \rightarrow \text{int}$ 
  - $\text{pre } \text{cptGele}(C, \text{id}) \text{ require } \text{id} \in \text{mPerso.keySet}() \wedge \text{estGele}(C, \text{id})$
- $\text{position} : [\text{GestionCombat}] \times \text{String} \rightarrow \text{Position}$ 
  - $\text{pre } \text{position}(C, \text{id}) \text{ require } \text{id} \in \text{mPerso.keySet}()$
- $\text{collisionGauche} : [\text{GestionCombat}] \times \text{String} \times \text{String} \rightarrow \text{boolean}$ 
  - $\text{pre } \text{collisionGauche}(C, \text{id1}, \text{id2}) \text{ require } \text{id1} \in \text{mPerso.keySet}() \wedge \text{id2} \in \text{mPerso.keySet}()$
- $\text{collision} : [\text{GestionCombat}] \times \text{String} \rightarrow \text{List}<\text{Personnage}>$ 
  - $\text{pre } \text{collision}(C, \text{id}) \text{ require } \text{id} \in \text{mPerso.keySet}()$

## 1.5 Constructors :

### 1.5.1

- $\text{init} : \text{int} \times \text{int} \times \text{int} \rightarrow [\text{GestionCombat}]$

## 1.6 Operators :

### 1.6.1

- $\text{gerer} : [\text{GestionCombat}] \times \text{Map} < \text{String}, \text{COMMANDE} > \rightarrow [\text{GestionCombat}]$

## 1.7 Observations :

### 1.7.1 [invariants]

- $\text{collision}(\text{C}, \text{id}) = \min$ 
  - $\text{tmp} = \emptyset, \forall \text{nomP} \in \text{mPerso.keySet}(),$
  - $\text{tmp} \cup \text{nomP}$  si  $\text{collisionGauche}(\text{C}, \text{id}, \text{nomP}) \parallel \text{collisionGauche}(\text{C}, \text{nomP}, \text{id})$
- $\text{collisionGauche}(\text{C}, \text{id1}, \text{id2}) = \min$ 
  - $\text{pos1} = \text{position}(\text{C}, \text{id1}), \text{pos2} = \text{position}(\text{C}, \text{id2}),$
  - $\text{Position}::\text{collision}(\text{pos1}, \text{pos2})$

### 1.7.2 [init]

- $\text{terrain}(\text{init}(\text{l}, \text{h}, \text{p})) = \text{Terrain}::\text{init}(\text{l}, \text{h}, \text{p})$
- $\text{mPerso}(\text{init}(\text{l}, \text{h}, \text{p})) =$ 
  - $\text{mPerso.put}(\text{"Alex"}, \text{Personnage}::\text{init}(\text{"Alex"}, 20, 51, 10, 100, 1664))$
  - $\text{mPerso.put}(\text{"Ryan"}, \text{Personnage}::\text{init}(\text{"Ryan"}, 25, 60, 12, 200, 1664))$
  - $\text{mPerso.put}(\text{"Slick"}, \text{Gangster}::\text{init}(\text{"Slick"}, 35, 80, 20, 250, 2000))$

– + 3 gangster lambda :

\* name = RandomName()

\* mPerso.put(name, Gangster::init(name,18,45,9,50,500))

• estFrappe(init(l,h,p),id)=false

• estGele(init(l,h,p),id)=false

• position

– position(init(l,h,p),"Alex") = Position::init(0, Terrain::profondeur(init(l,h,p))  
/ 2 + 1, 0, false)

– position(init(l,h,p),"Ryan") = Position::init(0, Terrain::profondeur(init(l,h,p))  
/ 2 - 1, 0, false)

– position(init(l,h,p),"Slick") = Position::init(Terrain::largeur(init(l,h,p))  
- 1, Terrain::profondeur(init(l,h,p)) / 2, 0, true)

– position(init(l,h,p), id) = Random sur Bloc Vide

### 1.7.3 [gerer]

• id n'appartient pas a keySet, creation d'un gangster

•  $\forall id \in mPerso.keySet(), mPerso(gerer(C, cmd)).get(id) =$

– si  $\neg estFrappe(gerer(C, cmd), id)$  alors  $mPerso(C).get(id)$

– sinon  $Personnage::retrait(mPerso(C).get(id), cpt)$

- \* avec  $\text{cpt}=0, \forall p \in \text{collision}(\text{C}, \text{id}), \text{cmd.get}(\text{Personnage}::\text{nom}(p))$   
 $= \text{FRAPPE}, \text{cpt} + \text{Personnage}::\text{force}(p)$
- $\forall \text{id} \in \text{mPerso.keySet}(), \text{estFrappe}(\text{gerer}(\text{C}, \text{cmd}), \text{id}) =$ 
  - $\text{collision}(\text{C}, \text{id}) \neq \emptyset \wedge (\exists p \in \text{collision}(\text{C}, \text{id}) \text{ tq } \text{cmd.get}(\text{Personnage}::\text{nom}(p))$   
 $= \text{FRAPPE})$
- $\forall \text{id} \in \text{mPerso.keySet}(), \text{estGele}(\text{gerer}(\text{C}, \text{cmd}), \text{id}) = (\text{cmd.get}(\text{id}) =$   
 $\text{FRAPPE}) \vee \text{estFrappe}(\text{gerer}(\text{C}, \text{cmd}), \text{id}) \vee \text{Chose}::\text{estPorte}(\text{mPerso}(\text{gerer}(\text{C},$   
 $\text{cmd})).\text{get}(\text{id})) \vee (\text{estGele}(\text{C}, \text{id}) \wedge \text{cptGele}(\text{C}, \text{id}) > 1)$
- $\forall \text{id} \in \text{mPerso.keySet}(), \text{cptGele}(\text{gerer}(\text{C}, \text{cmd}), \text{id}) =$ 
  - si  $(\text{cmd.get}(\text{id}) = \text{FRAPPE})$  alors 1
  - sinon si  $\text{estFrappe}(\text{gerer}(\text{C}, \text{cmd}), \text{id})$  alors 3
  - sinon  $\text{cpt}(\text{C}, \text{id})-1$
- $\text{mPerso.keySet}(), \text{position}(\text{gerer}(\text{C}, \text{cmd}), \text{id}) =$ 
  - si  $\text{estFrappe}(\text{gerer}(\text{C}, \text{cmd}), \text{id}),$ 
    - \* si  $(\exists p \in \text{collisionGauche}(\text{C}, \text{id}, p) \text{ tq } \text{cmd.get}(\text{Personnage}::\text{nom}(p))$   
 $= \text{FRAPPE})$  alors
      - $\text{Position}::\text{setX}(\text{position}(\text{C}, \text{id}), \min(\text{Position}::\text{x}(\text{position}(\text{C},$   
 $\text{id})) + 3, \text{Terrain}::\text{largeur}(\text{terrain}(\text{C}))))$
    - \* si  $(\exists p \in \text{collisionGauche}(\text{C}, p, \text{id}) \text{ tq } \text{cmd.get}(\text{Personnage}::\text{nom}(p))$   
 $= \text{FRAPPE})$  alors

- Position::setX(position(C, id), max(Position::x(position(C, id)) - 3, 0))
- si Chose::estPorte(mPerso(C).get(id)) alors
  - \*  $\exists p \in \text{mPerso}(C).\text{keySet}() \text{ tq } \text{Personnage}::\text{laChoseEquipee}(p) == \text{mPerso}(C).\text{get}(\text{id}) \wedge \text{cmd}.\text{get}(\text{Personnage}::\text{nom}(p)) \neq \text{JETER}$ 
    - Position::set(position(C, id), position(C, p))
  - \*  $\exists p \in \text{mPerso}(C).\text{keySet}() \text{ tq } \text{Personnage}::\text{laChoseEquipee}(p) = \text{mPerso}(C).\text{get}(\text{id}) \wedge \text{cmd}.\text{get}(\text{Personnage}::\text{nom}(p)) = \text{JETER}$ 
    - si Position::dirG(position(gerer(C, cmd), p)) alors Position::set(position(C, id), Position::x(max(position(C, p) - 5, 0)), Position::y(position(C, p)), 0)
    - sinon Position::set(position(C, id), Position::x(min(position(C, p) + 5, Terrain::largeur(terrain(C))), Position::y(position(C, p)), 0)
- si cmd.get(id) == DROITE
  - \* Position::setX(position(C, id), min( Position::x(position(C, id)) + 1, Terrain::largeur(terrain(C))))
  - \* Position::setDir(position(C, id), false);
- si cmd.get(id) == GAUCHE
  - \* Position::setX(position(C, id), max( Position::x(position(C, id)) - 1, 0))
  - \* Position::setDir(position(C, id), true);

```

- si cmd.get(id) == HAUT

    * Position::setY(position(C, id), min( Position::y(position(C,
      id)) + 1, Terrain::profondeur(terrain(C))))

- si cmd.get(id) == BAS

    * Position::setY(position(C, id), max( Position::y(position(C,
      id)) - 1, 0))

- si cmd.get(id) == SAUTER

    * Position::setZ(position(C, id), 1)

- sinon

    * Position::setZ(position(C, id), 0)

```