robot.h

```
//
// Created by nicolas-heig on 12/12/2022.
//
#ifndef INC_07_ROBOTS_ROBOT_H
#define INC_07_ROBOTS_ROBOT_H

#include <string>
#include "terrain.h"
#include <iostream>
#include <vector>
#include "game.h"

class Robot{
public:

    /** Fonction permettant d'afficher notre robot
     * ____________________________________________________________
     * @param os        : opérateur de flux
     * @param robot     : robot à afficher
     * ____________________________________________________________
     * @return          : retoune un flux d'affichage
     * @exception       : -
     */
    friend std::ostream& operator<< (std::ostream& os, const Robot& robot);

    /** Fonction permettant de détruie un Robot
     * ____________________________________________________________
     * @param vRobots   : vecteur contenant tous les robots
     * @param robot     : robot qui va tuer le robot au même emplacement
     * @param id        : id du robot à supprimer
     * @param terrain   : terrain pour l'affichage
     * ____________________________________________________________
     * @return          : -
     * @exception       : -
     */
    friend void detruireRobot(std::vector<Robot>& vRobots, Robot& robot, int id, Terrain& terrain);

    /** Constructeur par défaut de robot
     * ____________________________________________________________
     * @return          : -
     * @exception       : -
     */
    Robot();

    /** Constructeur de robot
     * ____________________________________________________________
     * @param terrain
     * ____________________________________________________________
     * @return          : -
     * @exception       : -
     */
    Robot(Terrain& terrain);

    Robot(int id, int posX, int posY);

    /**   Constructeur de copie
     * ____________________________________________________________
     * @param robot     : robot passé en copie
     * ____________________________________________________________
     * @return          : -
     * @exception       : -
     */
    Robot(const Robot& robot);
```

```
    /**   Opérateur d'affectation
     * ____________________________________________________________
     * @param robot     : robot passé en paramètre pour l'affectation
     * ____________________________________________________________
     * @return          : retourne un robot en référence
     * @exception       : -
     */
    Robot& operator=(const Robot& robot);

    //void deplacement(Robot& robot, Terrain& terrain);

    /**
     *
     * @param robot
     */

    /**   Fonction afin de déplacer le robot
    * ____________________________________________________________
    * @param terrain        : terrain afin de pouvoir positioner le tableau dans ce dernier
    * ____________________________________________________________
    * @return               : -
     * @exception           : -
     */
    void deplacer(Game game);


    /**   get l'id du robot
    * ____________________________________________________________
    * @return               : retourne l'id du robot
    * @exception           : -
    */
    int getId() const;


    /**   get la position en X du robot
    * ____________________________________________________________
    * @return               : retour de la position X
    * @exception           : -
    */
    int getPosX() const;


    /**   get la position en Y du robot
    * ____________________________________________________________
    * @return               : retour de la position Y
    * @exception           : -
    */
    int getPosY() const;


    /**
     *  Déconstructeur de la class Robot
     */
    /**   Déconstructeur de la class Robot
    * ____________________________________________________________
    * @return               : -
    * @exception            : -
    */
    ~Robot();

    Robot(int posX, int posY);
    bool operator() (int posX, int posY)


private:
```

```
    //id du robot
    const int id;

    //Position en x en y du robot
    int posX,
        posY;

    //nombre de robot total.
    static int nbreRobots;
    //prochain Id pour un robot
    static int prochainId;
};


#endif //INC_07_ROBOTS_ROBOT_H
```