

mai 20, 11 20:07

commentsauvegarde.cc

Page 1/4

```

/##### GrilleCellule.h #####
class GrilleCellule
{
    private:
        vector< vector<Cellule> > _grilleCellule;

        int _min_alive, _max_alive;
        int _min_dead, _max_dead;

        ...etc

        void lireGrilleCellule( fstream & f );
        void ecrireGrilleCellule( fstream & f );
};

/##### GrilleCellule.h #####

... blablabla

/**
 * \fn GrilleCellule::lireGrilleCellule( fstream & f )
 * \brief modifie les Cellules a partir d'un flux
 * \param f un flux qui contient les informations necessaire
 * \return rien
 */
void GrilleCellule::lireGrilleCellule( fstream & f )
{
    int ti, tj;
    float x,y;
    int val;
    state e;
    health s;
    vector<Cellule> tmp;

    f >> _min_alive >> _max_alive >> _min_dead >> _max_dead;
    f >> _aleaCellule >> _aleaResist;
    f >> ti >> tj;
    f >> _wC >> _hC;

    _grilleCellule.clear();
    for ( int i = 0; i < ti; i++ )
    {
        for ( int j = 0; j < tj; j++ )
        {
            f >> x >> y;

            f >> val;
            e = getState(val);
            f >> val;
            s = getHealth(val);

            tmp.push_back( Cellule(x,y,e,s) );
        }
        _grilleCellule.push_back( tmp );
        tmp.clear();
    }
}

/**

```

vendredi mai 20, 2011

mai 20, 11 20:07

commentsauvegarde.cc

Page 2/4

```

* \fn GrilleCellule::ecrireGrilleCellule( fstream & f )
* \brief charge les Cellules a partir d'un flux
* \param f un flux qui contiendra les informations necessaire
* \return rien
*/
void GrilleCellule::ecrireGrilleCellule( fstream & f )
{
    int ti, tj;
    getTailleGrille(ti, tj);

    f << _min_alive << " " << _max_alive << " " << _min_dead << " " << _max_d
ead << endl;
    f << _aleaCellule << " " << _aleaResist << endl;
    f << ti << " " << tj << endl;
    f << _wC << " " << _hC << endl;

    for ( int i = 0; i < ti; i++ )
        for ( int j = 0; j < tj; j++ )
        {
            f << _grilleCellule[i][j].getX() << " " << _grilleCellul
e[i][j].getY() << endl;
            f << int(_grilleCellule[i][j].getEtat()) << " " << int(_
grilleCellule[i][j].getSante()) << endl;
        }
}

/##### GroupeVirus.h #####

class GroupeVirus
{
    private:
        float _wVirus;
        float _hVirus;
        int _nbEnfant;
        float _dureVie, _dureIncub;
        vector<Virus> _groupeVirus;

    public:

        ... etc

        void lireGroupeVirus( fstream & f );
        void ecrireGroupeVirus( fstream & f );
};

/##### GroupeVirus.cc #####

... blabla

/**
 * \fn GroupeVirus::lireGroupeVirus( fstream & f )
 * \brief modifie les Virus a partir d'un flux
 * \param f un flux qui contient les informations necessaire
 * \return rien
 */
void GroupeVirus::lireGroupeVirus( fstream & f )
{
    int ti;
    bool t;
    float p, ip, x, y;

```

commentsauvegarde.cc

1/2

mai 20, 11 20:07

commentsauvegarde.cc

Page 3/4

```

f >> ti;
f >> _wVirus >> _hVirus;
f >> _nbEnfant >> _dureVie >> _dureIncub;

_groupeVirus.clear();

for( int i=0; i<ti; i++ )
{
    f >> x >> y >> t >> p >> ip;

    _groupeVirus.push_back( Virus(x,y) );
    _groupeVirus[i].setTarget(t);
    _groupeVirus[i].setTimePassed(p);
    _groupeVirus[i].setTimeIncPassed(ip);
}

/**
 * \fn GroupeVirus::lireGroupeVirus( fstream & f )
 * \brief charge les Virus a partir d'un flux
 * \param f un flux qui contiendra les informations necessaire
 * \return rien
 */
void GroupeVirus::ecrireGroupeVirus( fstream & f )
{
    int ti = getSizeGroupe();
    f << ti << endl;
    f << _wVirus << " " << _hVirus << endl;
    f << _nbEnfant << " " << _dureVie << " " << _dureIncub << endl;

    for( int i=0; i<ti; i++ )
    {
        f << _groupeVirus[i].getX() << " " << _groupeVirus[i].getY() <<
endl;
        f << _groupeVirus[i].getTarget() << " " << _groupeVirus[i].getTi
mePassed() << " " << _groupeVirus[i].getTimeIncPassed() << endl;
    }
}

##### GameSave.cc #####

/**
 * \fn GameSave::open(string nom_sauvegarde, GrilleCellule* grille, GroupeVirus*
groupe, bool & s)
 * \brief ouvre une sauvegarde
 * \param grille GrilleCellule Ã modifier
 * \param groupe GroupeVirus Ã modifier
 * \param s boolÃen definissant l'etat du jeu a modifier
 * \return rien
 */
void GameSave::open(string nom_sauvegarde, GrilleCellule* grille, GroupeVirus* g
roupe)
{
    fstream f;
    f.open( nom_sauvegarde.data(), ios::in );

    grille->lireGrilleCellule(f);
    groupe->lireGroupeVirus(f);
}

/**
 * \fn GameSave::save(string nom_sauvegarde, GrilleCellule* grille, GroupeVirus*

```

mai 20, 11 20:07

commentsauvegarde.cc

Page 4/4

```

groupe, const bool & s)
 * \brief sauvegarde le jeu en cours
 * \param grille GrilleCellule Ã enregistrer
 * \param groupe GroupeVirus Ã enregistrer
 * \param s boolÃen definissant l'etat du jeu Ã enregistrer
 * \return rien
 */
void GameSave::save(string nom_sauvegarde, GrilleCellule* grille, GroupeVirus* g
roupe)
{
    fstream f;
    f.open( nom_sauvegarde.data(), ios::out );

    grille->ecrireGrilleCellule(f);
    groupe->ecrireGroupeVirus(f);

    f.close();
}

```