

May 31, 16 7:55	Model.h	Page 1/2
<pre> #ifndef _MODEL_ #define _MODEL_ #include <vector> #include <time.h> #include <SFML/Graphics.hpp> #include "Constantes.h" #include "AutonomousElement.h" #include <fstream> /* * ----- * "THE BEER-WARE LICENSE" (Revision 42): * Antoine BOULANGAM-^I & Pierre_Elliot CABRERA wrote this file. As long as you retain this notice you * can do whatever you want with this stuff. If we meet some day, and you think * this stuff is worth it, you can buy me a beer in return * ----- */ class MovableElement; class Ninja; class Model { private: int _w, _h; int _hp; bool _damaged; int _score; int _money; bool _invincibility; DIFFICULTY _difficulty; bool _introduction; bool _menu; bool _mort; bool _highscore; bool _menuHighscore; Ninja * _ninja; std::vector<AutonomousElement *> _elements; std::vector<AutonomousElement *> _elementsDeleted; std::vector<const AutonomousElement *> _newElements; std::vector<const AutonomousElement *> _newElementsDeleted; sf::Clock _clockBonus; sf::Clock _clockObstacle; sf::Time _repopTimeObstacle; sf::Time _repopTimeBonus; sf::Clock _clockInvincibility; sf::Clock _clockFly; public: //Constructeur et destructeur Model(int w, int h); ~Model(); //Accesseurs void getNinjaPosition(int &x, int &y) const; void getNinjaDimension(int &w, int &h) const; NINJA_STATUS getNinjaStatus() const; void getNinjaSpeedX(int &mvtX) const ; void getNinjaSpeedY(int &mvtY) const ; bool getNinjaToStopX() const; bool getNinjaToStopY() const; </pre>		

May 31, 16 7:55	Model.h	Page 2/2
<pre> int getNinjaToChangeDirection() const; int getHp(); int getScore(); std::vector<const AutonomousElement *> getNewAutonomousElements() const; std::vector<const AutonomousElement *> getNewElementsDeleted() const; std::vector<AutonomousElement *> getAutonomousElements() const; std::vector<AutonomousElement *> getElementsDeleted() const; bool getDamaged(); bool getMenu() const; bool getDead() const; bool getInvincibility() const; LANGUAGE getLanguage() const; int getMoney() const; bool getHighscore() const; bool getIntroduction() const; bool getMenuHighscore() const; //Mutateurs void setNinjaStatus(NINJA_STATUS value); void setNinjaDirection(bool forward); void setNinjaToStopX(bool toStop); void setNinjaToStopY(bool toStop); void setNinjaToChangeDirection(NINJA_CHANGE_DIRECTION newDirection); void setNinjaXBeginJump(int value); void setNinjaYBeginJump(int value); void setNinjaReduction(bool toReduce); void setMenu(bool menu); void setDead(bool mort); void setHighscore(bool highscore); void setIntroduction(bool introduction); void setMoney(int money); void setMenuHighScore(bool menuHighscore); //MÃ©thodes void addScore(int value); void changeLife(int gain = 1); void addSpeed(); void moveNinjaX(bool forward); void moveNinjaY(bool up); void stopNinja(bool abscissa); void nextStep(); void addElement(bool obstacle); void clearNewElements(); void clearNewElementsDeleted(); void collisionNinja(); void changeDamaged(); }; #endif </pre>		