```
#ifndef _MODEL_
#define _MODEL_
#include "Ball.h"
#include <SFML/Graphics.hpp>
#include <vector>
#include "GraphicElement.h"
#include <math.h>
#include "MovableElement.h"
#include "Ball.h"
#include "GraphicElement.h"
#include "Obstacles.h"
#include "Score.h"
#include "AnimatedGraphicElement.h"

const int SCREEN_WIDTH = 800;
const int SCREEN_HEIGHT = 600;

class Model {
private:
    int _w, _h;
    Ball *_ball;
    std::vector < MovableElement * > _elements;
    std::vector<const MovableElement * > _new_elements;
    std::vector<MovableElement*> _elementsTruck;
    GraphicElement *_buttonPlay, *_buttonQuit;
    float x, y;
    Obstacles *_obstacle;
    Score *_score;
    sf::Vector2i limite;
    sf::Clock clock;
    sf::Clock clockbonus;
    sf::Time time=clock.getElapsedTime();
    sf::Time timeB = clock.getElapsedTime();
    int val = 1500;
    int bonusshow = 6000;
    int _clockDelay = 1000;
    int _clockBonus = 1000;

public:

    Model(int w, int h);
    ~Model();

    void nextStep();
    void getBallPosition(int&x, int&y);
    void getBallDimension(int&w, int&h);
    void moveBall(bool left);
    void getXB(int &x);
    void getYB(int &y);

    std::string writeScore(int score);
    int getScore();
    void setScore(int s);
    int getVie();
    void setVie(int nb);

    std::vector < const MovableElement * > getNewMovableElements() const;
    std::vector < MovableElement * > getMovableElements() const;
    void addElement(std::string type);
    void jumpBall();

    bool collision(MovableElement *ball, MovableElement *obstacle);
    void setPositionObstacle(int x, int y);

    void resultatCollision(MovableElement *obstacle, int type);
    void resultatCollisionBonus(MovableElement *bonus, int type);

    void bestScores();
```



```
    int jumpSpeed = 60;
};

#endif
```