

```
1: //Authors: Joseph Calles and Tharith Sovann
2:
3: #ifndef BODY_HPP
4: #define BODY_HPP
5:
6: // include SFML libraries
7: #include <SFML/Graphics.hpp>
8: #include <SFML/Window.hpp>
9: #include <SFML/System.hpp>
10: #include <SFML/Audio.hpp>
11:
12: // include directives
13: #include <iostream>
14: #include <string>
15: #include <memory>
16: #include <vector>
17: #include <cmath>
18:
19: using namespace std;
20:
21: class Body : public sf::Drawable {
22: public:
23:     Body():_x_position(0), _y_position(0),
24:           _x_velocity(0), _y_velocity(0), _mass(0) {}
25:
26:     Body(double x, double y, double xv,
27:           double yv, double m, string filename): _x_position(x), _y_position(y)
28:     ,
29:           _x_velocity(xv), _y_velocity(y
30: v),
31:           _mass(m), _filename(filename)
32: {}
33:
34:     friend istream& operator>>(istream& input, Body& x);
35:     friend ostream& operator<< (ostream& out, Body& body);
36:
37:     void set_radius(double radius){ _universe_radius = std::make_shared<
38: double>(radius); }
39:     void set_radius(shared_ptr<double> radius){ _universe_radius = radiu
40: s; }
41:
42:     void set_big_G(double G){ _big_G = make_shared<double>(G); }
43:
44:     void set_window_scale(sf::Vector2u window_size){ _window_size = wind
45: ow_size; }
46:     sf::Vector2u get_window_size(void) { return _window_size; }
47:
48:     void set_x_vel(double xvel) { _x_position = xvel; }
49:     void set_y_vel(double yvel) { _y_position = yvel; }
50:
51:     double calc_y_force(Body& other_planet);
52:     double calc_x_force(Body& other_planet);
53:
54:     void set_x_force(double xforce) { _x_force = xforce; }
55:     void set_y_force(double yforce) { _y_force = yforce; }
56:
57:     shared_ptr<double> get_radius(void){ return _universe_radius; }
58:     shared_ptr<double> get_G(void) { return _big_G; }
59:
60:     double get_mass(void) { return _mass; }
61: }
```

```
56:         void step(double delta_seconds);
57:         void set_new_position(void);
58:     private:
59:
60:         void draw(sf::RenderTarget& target, sf::RenderStates state) const
61:         { target.draw(_sprite, state); } // draw function overrider
62:
63:         double _x_position;
64:         double _y_position;
65:
66:         double _x_velocity;
67:         double _y_velocity;
68:
69:         double _x_accel;
70:         double _y_accel;
71:
72:         double _x_force;
73:         double _y_force;
74:
75:         double _mass;
76:         string _filename;
77:
78:         shared_ptr<double> _universe_radius;
79:         shared_ptr<double> _big_G;
80:
81:
82:         sf::Vector2u _window_size;
83:         sf::Texture _texture;
84:         sf::Image _image;
85:         sf::Sprite _sprite;
86: };
87:
88: #endif /* BODY_HPP */
```