```
Body.cpp
                Thu Mar 01 22:52:26 2018
                                                 1
    1: //
    2: //
           Body.cpp
    3: //
           ps3a
    4: //
    5: //
           Created by Jingxian Shi on 2/19/18.
           Copyright © 2018 Jingxian Shi. All rights reserved.
    6: //
    7: //
    8:
    9: #include "Body.hpp"
   10:
   11: Body::Body()
   12: {
   13:
   14: }
   15:
   16: Body::Body (double xpos, double ypos, double xvel, double yvel, double mass, std::
string file_name)
   17: {
   18:
           _{xpos} = xpos;
   19:
           _ypos = ypos;
           _xvel = xvel;
   20:
           _yvel = yvel;
   21:
           _mass = mass;
   22:
           _image_file = file_name;
   23:
   24:
   25:
           _texture.loadFromFile(_image_file);
   26:
           _sprite.setTexture(_texture);
   27:
   28: }
   29:
   30: Body::~Body()
   31: {
   32:
   33: }
   34:
   35: std::ostream& operator<<(std::ostream& out, Body& body)
   36: {
   37:
           out << body._xpos << " "
           << body._ypos << " "
   38:
           << body._xvel << " "
   39:
           << body._yvel << " "
   40:
           << body._mass << " "
   41:
   42:
           << body._image_file;</pre>
   43:
   44:
           return out;
   45: }
   47: std::istream& operator>>(std::istream& in, Body& body)
   48: {
   49:
           in >> body._xpos >> body._ypos >> body._xvel >> body._yvel >> body._mass >> b
ody._image_file;
   50:
           body._texture.loadFromFile(body._image_file);
   51:
           body._sprite.setTexture(body._texture);
   52:
   53:
           return in;
   54: }
   55:
   56: void Body::draw(sf::RenderTarget& target, sf::RenderStates states) const
   57: {
   58:
           sf::Sprite image = _sprite;
   59:
   60:
           image.setPosition(_window_size/2.0 + _xpos * _window_size * 0.5 / _univ_size,
 _window_size/2.0 - _ypos * _window_size * 0.5 / _univ_size);
   61:
   62:
           target.draw(image, states);
```

```
Thu Mar 01 22:52:26 2018 2
Body.cpp
   63: }
   64:
   65: void Body::step(double seconds, double force_x, double force_y)
   66: {
   67:
            double accel_x = force_x / _mass;
   68:
           double accel_y = force_y / _mass;
   69:
          _xvel += accel_x * seconds;
_yvel += accel_y * seconds;
   70:
   71:
   72:
          _xpos += _xvel * seconds;
_ypos += _yvel * seconds;
   73:
   74:
   75: }
```

```
Body.hpp
    1: //
    2: //
           Body.hpp
    3: //
          ps3a
    4: //
    5: // Created by Jingxian Shi on 2/19/18.
           Copyright © 2018 Jingxian Shi. All rights reserved.
    6: //
    7: //
    8:
    9: #ifndef Body_hpp
   10: #define Body_hpp
   11:
   12: #include <stdio.h>
   13: #include <iostream>
   14: #include <string>
   15: #include <SFML/Graphics.hpp>
   16:
   17: class Body : public sf::Drawable
   18: {
   19: public:
   20:
           Body();
   21:
           Body (double xpos, double ypos, double xvel, double yvel, double mass, std::st
ring file_name);
   22:
           ~Body();
   23:
           void setWindowSize(int window_size) {_window_size = window_size;}
   24:
          void setUnivSize(double univ_size) {_univ_size = univ_size;}
   25:
           double getUnivSize() {return _univ_size;}
   26:
          void step(double seconds, double force_x, double force_y);
   27:
   28:
           double getPosX() {return _xpos;}
   29:
           double getPosY() {return _ypos;}
   30:
           double getVelX() {return _xvel;}
   31:
           double getVelY() {return _yvel;}
   32:
           double getMass() {return _mass;}
   33:
   34:
           friend std::ostream& operator << (std::ostream& out, Body& body);
   35:
           friend std::istream& operator>>(std::istream& in, Body& body);
   36:
   37: private:
   38:
           virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const;
   39:
   40:
           double _univ_size;
   41:
           int _window_size;
   42:
           double _xpos, _ypos, _xvel, _yvel, _mass;
           std::string _image_file;
   43:
   44:
           sf::Texture _texture;
   45:
           sf::Sprite _sprite;
   46: };
   47:
   48:
   49: #endif /* Body_hpp */
   50:
```

Thu Mar 01 22:17:28 2018