```
#include "NBody.hpp"
//.000000002
const double SCALE = (3.5e+11);//This number negates the e+10 in the x position.
const double G = (6.67e-11);
void Body::draw(sf::RenderTarget &target, sf::RenderStates states) const{
  sf::Image image;
  sf::Texture texture;
  sf::Sprite sprite;

  //std::cout << _position.x << _position.y << _filename << std::endl;

  if(!image.loadFromFile(_filename)){
      std::cout << "ERROR: could not load image from file" << std::endl;
      return;
    }

    texture.loadFromFile(_filename);
    sprite.setTexture(texture);

    //need to multiply the x position by SCALE so the planets are not off the screen
    sprite.setPosition((_position.x/SCALE) * 500 + target.getSize().x/2,(_position.y/SCALE) * 500 + target.getSize().y/2);
//x position / universe size * window size
    target.draw(sprite);

}

std::istream& operator>>(std::istream& in, Body& body){

  in >> body._position.x  >> body._position.y >> body._velocity.x >> body._velocity.y >> body._mass >> body._filename;

  return in;

}

Body::Body(double xCoord, double yCoord, double xVelocity, double yVelocity, double mass, std::string fileName){
  //setting vars to specifications
  _position.x = xCoord;
  _position.y = yCoord;
  _velocity.x = xVelocity;
  _velocity.y = yVelocity;
  _mass = mass;
  _filename = fileName;

}
Body::Body(){

}

void Body::step(double t_time){


  sf::Vector2f distance;
  sf::Vector2f o_accel = getAccel();
  sf::Vector2f oldVel = getVel();
  sf::Vector2f newVel;
  sf::Vector2f endVel;
  sf::Vector2f oldPosition = getPosition();
  sf::Vector2f newPosition;

  distance.x = t_time * oldVel.x;
  distance.y = t_time * oldVel.y;

  newPosition.x = distance.x + oldPosition.x;
  newPosition.y = distance.y + oldPosition.y;

  newVel.x = o_accel.x * t_time;
  newVel.y = o_accel.y * t_time;

  endVel.x = newVel.x + oldVel.x;
  endVel.y = newVel.y + oldVel.y;


  setPosition(newPosition);

  setVel(endVel);


}


  void Body::setNetforce(sf::Vector2f sNetforce){
    _netforce = sNetforce;
  }
  sf::Vector2f Body::getNetforce(){
    return _netforce;
  }

  void Body::setNumPlanets(int sNoplanets){
    _numberOfPlanets = sNoplanets;
  }
  int Body::getNumPlanets(){
    return _numberOfPlanets;
  }

  void Body::setupdatePos(sf::Vector2f sUpdatepos){
    _updatedPosition = sUpdatepos;
  }
  sf::Vector2f Body::getupdatePos(){
    return _updatedPosition;
  }



  void Body::setPosition(sf::Vector2f sPosition){
   _position = sPosition;
  }

 sf::Vector2f Body::getPosition(){
  return _position;
 }

 void Body::setVel(sf::Vector2f sVelocity){
  _velocity = sVelocity;
 }

 sf::Vector2f Body::getVel(){
  return _velocity;
 }

  void Body::setAccel(sf::Vector2f sAccel){
   _acceleration = sAccel;
  }

sf::Vector2f Body::getAccel(){
  return _acceleration;
}

void Body::setMass(double sMass){
  _mass = sMass;
}

double Body::getMass(){
  return _mass;
 }

void Body::setFilename(std::string sFile){
    _filename = sFile;
  }

 std::string Body::getFilename(){
  return _filename;
 }

 Body::~Body(){

 }

//in main, before we make a new Body, we read in the file name that has the proper characteristics, save them into multiple vars or strings, then feed them into the Body constructor
```