```
    1: /**
    2: * CelestialBody.cpp - an implementation to create each celestial body
    3: * 1 at a time, and also place them into the correct location
    4: * for drawing. plus draw each one individually, and overode >> operator
    5: *
    6: * Date 2/14/22 - 2/22/22
    7: *
    8: * Created by: Anson Cheang
    9: *
   10: */
   11:
   12: #include "CelestialBody.h"
   13: #include <SFML/System.hpp>
   14: #include <SFML/Window.hpp>
   15: #include <SFML/Graphics.hpp>
   16: #include <string>
   17: #include <cstdlib>
   18: #include <iostream>
   19:
   20: using namespace std;
   21:
   22: CelestialBody::CelestialBody(double val)
   23: {
   24:     scale = val;
   25:     XPosition = 0;
   26:     YPosition = 0;
   27:     XVelocity = 0;
   28:     YVelocity = 0;
   29:     Mass = 0;
   30:     filename = "";
   31: }
   32:
   33:
   34: void CelestialBody::createImage()
   35: {
   36:     if(!image.loadFromFile(filename))
   37:     {
   38:         exit(-1);
   39:     }
   40:
   41:     texture.loadFromImage(image);
   42:
   43:     sprite.setTexture(texture);
   44:     sf::Vector2u size = image.getSize();
   45:     sprite.setOrigin(static_cast<int>(size.x)/2, static_cast<int>(size.y)
/2);
   46:     sprite.setPosition(sf::Vector2f(XPosition*scale + 350, YPosition*scal
e + 350));
   47: }
   48:
   49: CelestialBody::CelestialBody(double posX, double posY, double Xvel, doubl
e Yvel, double Imass, string _filename)
   50: {
   51:     XPosition = posX;
   52:     YPosition = posY;
   53:     XVelocity = Xvel;
   54:     YVelocity = Yvel;
   55:     Mass = Imass;
   56:     filename = _filename;
   57:
   58:     if(!image.loadFromFile(filename))
   59:     {
   60:         exit(-1);
   61:     }
   62:
```

```
   63:     texture.loadFromImage(image);
   64:
   65:     sprite.setTexture(texture);
   66:     sf::Vector2u size = image.getSize();
   67:     sprite.setOrigin(static_cast<int>(size.x)/2, static_cast<int>(size.y)
/2);
   68:     sprite.setPosition(sf::Vector2f(posX, posY));
   69: }
   70:
   71: void CelestialBody::draw(sf::RenderTarget& target, sf::RenderStates state
s) const
   72: {
   73:     target.draw(sprite, states);
   74: }
   75:
   76: istream& operator>>(istream& instream, CelestialBody& planet)
   77: {
   78:     instream >> planet.XPosition >> planet.YPosition >> planet.XVelocity
>> planet.YVelocity >> planet.Mass >> planet.filename;
   79:     return instream;
   80: }
   81:
   82: void CelestialBody::setPos(sf::Vector2f Pos)
   83: {
   84:     XPosition = Pos.x;
   85:     YPosition = Pos.y;
   86: }
   87:
   88: void CelestialBody::setVel(sf::Vector2f Vel)
   89: {
   90:     XVelocity = Vel.x;
   91:     YVelocity = Vel.y;
   92: }
   93:
   94: double CelestialBody::getXPos()
   95: {
   96:     return XPosition;
   97: }
   98:
   99: double CelestialBody::getYPos()
  100: {
  101:     return YPosition;
  102: }
  103:
  104: double CelestialBody::getMass()
  105: {
  106:     return Mass;
  107: }
  108:
  109: void CelestialBody::setImagePos()
  110: {
  111:     //double CX = (Pos.x - XPosition) * scale;
  112:     //double CY = (Pos.y - YPosition) * scale;
  113:     sprite.setPosition(sf::Vector2f(XPosition*scale + 350, YPosition*scal
e + 350));
  114:     //cout << XPosition*scale + 350 << ", " << YPosition*scale + 350 << e
ndl;
  115:     //XPosition = Pos.x;
  116:     //YPosition = Pos.y;
  117: }
  118:
  119: double CelestialBody::getXVel()
  120: {
  121:     return XVelocity;
  122: }
```

```
  123:
  124: double CelestialBody::getYVel()
  125: {
  126:     return YVelocity;
  127: }
  128:
  129:
  130: ostream& operator<<(ostream& out, CelestialBody planet)
  131: {
  132:     out << planet.XPosition << "  " << planet.YPosition << "  " << planet
.XVelocity << "  "
  133:         << planet.YVelocity << "  " << planet.Mass << "  " << planet.file
name << endl;
  134:     return out;
  135: }
```