```
 1: #include <sstream>
 2:
 3: #include "planets.cpp"
 4:
 5: #define BACKGROUND "background.jpg"
 6:
 7: Vector2u WINSIZE, backSize;
 8:
 9: int main(int argc, char* argv[]) {
10:     int N;
11:     double R;
12:     double T = atoi(argv[1]);
13:     double deltaT = atoi(argv[2]);
14:     vector<shared_ptr<Body>> bodies;
15:     double dx = 0,
16:            dy = 0,
17:            r = 0,
18:            f = 0,
19:            fX = 0,
20:            fY = 0;
21:
22:     RenderWindow window(VideoMode(800, 800), "NBody");
23:
24:     Sprite background;
25:     Texture texture;
26:     if (!texture.loadFromFile(BACKGROUND)) {
27:         cout << "Failed to load background" << endl;
28:         return EXIT_FAILURE;
29:     }
30:
31:     Music music;
32:     if (!music.openFromFile("HEYYEYAAEYAAAEYAEYAA.ogg")) {
33:         cout << "Failed to load music" << endl;
34:         return EXIT_FAILURE;
35:     }
36:     music.play();
37:
38:     Font font;
39:     font.loadFromFile("digital-7 (mono).ttf");
40:     Text text;
41:     text.setFont(font);
42:     text.setPosition(0, 0);
43:     text.setCharacterSize(24);
44:     stringstream timer;
45:
46:     backSize = texture.getSize();
47:     WINSIZE = window.getSize(); //gets the window size
48:
49:     double xScale = (double) WINSIZE.x / backSize.x;
50:     double yScale = (double) WINSIZE.y / backSize.y;
51:
52:     background.setTexture(texture);
53:     background.setScale(xScale, yScale);
54:
55:     window.setFramerateLimit(60);
56:
57:     cin >> N;
58:     cin >> R;
59:     for (int i = 0; i < N; i++) {
60:         shared_ptr<Body> ptrBody(new Body());
61:         cin >> *ptrBody;
```

```
   62:         ptrBody->scale(WINSIZE, R);
   63:         bodies.push_back(ptrBody);
   64:     }
   65:
   66:     Clock clock;
   67:     clock.restart();
   68:
   69:     while (window.isOpen())
   70:     {
   71:         Time ElapsedTime = clock.getElapsedTime();
   72:         double timePassed = ElapsedTime.asSeconds();
   73:         timer.str(string());
   74:         timePassed *= deltaT;
   75:         timer << "Time passed in seconds: " << timePassed;
   76:         text.setString(timer.str().c_str());
   77:         Event event;
   78:         while (window.pollEvent(event))
   79:         {
   80:             if (event.type == Event::Closed || Keyboard::isKeyPressed(Keyboa
rd::Escape))
   81:                 window.close();
   82:         }
   83:
   84:         window.clear();
   85:         window.draw(background);
   86:         window.draw(text);
   87:
   88:         for (unsigned int i = 0; i < bodies.size(); i++) {
   89:             fX = 0;
   90:             fY = 0;
   91:             for (unsigned int j = 0; j < bodies.size(); j++) {
   92:                 if (i != j) {
   93:                     dx = bodies[j]->x - bodies[i]->x;
   94:                     dy = bodies[j]->y - bodies[i]->y;
   95:                     r = getRadius(dx, dy);
   96:                     f = getForce(bodies[i]->getM(), bodies[j]->getM(), r);
   97:                     fX += dirF(dx, f, r);
   98:                     fY += dirF(dy, f, r);
   99:                 }
  100:             }
  101:             bodies[i]->F.x = fX;
  102:             bodies[i]->F.y = fY;
  103:             bodies[i]->time(deltaT);
  104:             bodies[i]->move();
  105:             window.draw(*bodies[i]);
  106:         }
  107:
  108:         if (timePassed > T || Keyboard::isKeyPressed(Keyboard::Escape) || ev
ent.type == Event::Closed) {
  109:             for (unsigned int i = 0; i < bodies.size(); i++) {
  110:                 cout << " " << bodies[i]->x;
  111:                 cout << " " << bodies[i]->y;
  112:                 cout << " " << bodies[i]->xVel;
  113:                 cout << " " << bodies[i]->yVel;
  114:                 cout << " " << bodies[i]->mass;
  115:                 cout << " " + bodies[i]->img_file << endl;
  116:             }
  117:         }
  118:
  119:         window.display();
  120:     }
```

```
  121:     return 0;
  122: }
```