```
using System;
using System.Linq;
using System.Collections.Generic;
using System.Threading;
using System.Threading.Tasks;
using System.IO;
using Structures;
using static Program.Constants;
using Gtk;
using Gdk;
using Cairo;
using Graphics;
using static Program.Program;
namespace Program {
    static class Input {
        private static bool canMove = false;
        private static Vector3 rootPos = null;
        private static Vector3 rootAngle = null;
        private static readonly double MOUSE_SENSITIVITY = 1;
        private static readonly double SCROLL_SENSITIVITY = 1.1;
        private static readonly double TIME_SENSITIVITY = 1.2;
        private static readonly double RADIUS_SENSITIVITY = 1.1;
        private static readonly int LINE_SENSITIVITY = 5;
        private static double focal_length = -1;
        [GLib.ConnectBefore]
        public static void OnKeyPress(object sender, KeyPressEventArgs args) {
                if (args.Event.Key == Gdk.Key.f) {
                        if (Program.activesys == null) return;
                            else {
                    Program.activesys.IterateCenter();
                    Program.sys_view.ClearPaths();
                }
                args.RetVal = true;
            } else if (args.Event.Key == Gdk.Key.r) {
                double d = Vector3.Magnitude
(Program.sys_view.camera.position);
                Program.sys_view.camera = new Camera(d,Vector3.zero);
            } else if (args.Event.Key == Gdk.Key.l) {
                canMove = !canMove;
                if (!canMove) {
                    rootPos = null;
                }
            } else if (args.Event.Key == Gdk.Key.Up) {
                Program.sys_view.radius_multiplier *= RADIUS_SENSITIVITY;
            } else if (args.Event.Key == Gdk.Key.Down) {
                Program.sys_view.radius_multiplier /= RADIUS_SENSITIVITY;
            } else if (args.Event.Key == Gdk.Key.Right) {
                Program.activesys.Stop();
                Program.timestep *= TIME_SENSITIVITY;
                Program.activesys.StartAsync(step: Program.timestep);
            } else if (args.Event.Key == Gdk.Key.Left) {
                Program.activesys.Stop();
                Program.timestep /= TIME_SENSITIVITY;
                Program.activesys.StartAsync(step: Program.timestep);
            } else if (args.Event.Key == Gdk.Key.Page_Down) {
                // don't make it smaller than 0
                if (Program.sys_view.line_max >= LINE_SENSITIVITY) {
                    Program.sys_view.line_max -= LINE_SENSITIVITY;
                }
            } else if (args.Event.Key == Gdk.Key.Page_Up) {
                Program.sys_view.line_max += LINE_SENSITIVITY;
            } else if (args.Event.Key == Gdk.Key.Escape) {
                Program.sys_view.Stop();
                Program.activesys.Stop();
                Program.mainWindow.Destroy();

                var menu = new UI.Menu();
                var data = new UI.SaveData() {
                    bodies = ((IEnumerable<Body>)Program.activesys).ToList(),
                    centers = Program.CustomCenters,
                    timestep = Program.timestep,
                    radius_multiplier = Program.sys_view.radius_multiplier,
                    line_max = Program.sys_view.line_max
                };
                menu.temp_savedata = data;
                menu.loadButton.Click();
            } else if (args.Event.Key == Gdk.Key.q) {
                Program.sys_view.camera = new Camera(Vector3.Magnitude(Program.sys_view.camera.position)*SCROLL_SENSITIVITY,Program.sys_view.camera.angle);
            } else if (args.Event.Key == Gdk.Key.w) {
                Program.sys_view.camera = new Camera(Vector3.Magnitude(Program.sys_view.camera.position)/SCROLL_SENSITIVITY,Program.sys_view.camera.angle);
            } else if (args.Event.Key == Gdk.Key.c) {
                if (focal_length == -1) {
                    Console.WriteLine("hi");
                    focal_length = Vector3.Magnitude(Program.sys_view.camera.position);
                    Program.sys_view.camera = new Camera(1000*AU,Program.sys_view.camera.angle);
                    //Program.sys_view.ClearPaths();
                    //Program.sys_view.Redraw();
                } else {
                    Console.WriteLine("hi2");
                    Program.sys_view.camera = new Camera(focal_length,Program.sys_view.camera.angle);
                    //Program.sys_view.Redraw();
                    focal_length = -1;
                }
            }
            }
        [GLib.ConnectBefore]
        public static void OnMouseMovement(Object sender, MotionNotifyEventArgs args) {
            if (canMove) {
                if (rootPos == null || rootAngle == null ) {
                    rootPos = new Vector3(args.Event.X,args.Event.Y,0);
                    rootAngle = Program.sys_view.camera.angle;
                } else {
                    double d = Vector3.Magnitude(Program.sys_view.camera.position);
                    Program.sys_view.camera = new Camera(d,rootAngle + deg*MOUSE_SENSITIVITY* new Vector3(rootPos.y - args.Event.Y,0,args.Event.X - rootPos.x));
                } args.RetVal = true;
            }
        }
        [GLib.ConnectBefore]
        public static void OnScrollMovement(Object sender, ScrollEventArgs args) {
            if (args.Event.Direction == Gdk.ScrollDirection.Up) {
                Program.sys_view.bounds_multiplier /= SCROLL_SENSITIVITY;
                Program.sys_view.camera = new Camera(Vector3.Magnitude(Program.sys_view.camera.position)/SCROLL_SENSITIVITY,Program.sys_view.camera.angle);
            } else if (args.Event.Direction == Gdk.ScrollDirection.Down) {
                Program.sys_view.bounds_multiplier *= SCROLL_SENSITIVITY;
                Program.sys_view.camera = new Camera(Vector3.Magnitude(Program.sys_view.camera.position)*SCROLL_SENSITIVITY,Program.sys_view.camera.angle);

            }
        }
        }
    }
```