# The Simulation

The physics simulation provides the user the ability to create primitive physics objects and see the forces being applied (e.g. gravity) and collision reactions inside the scene. A selection of Circles, Boxes, Planes, and Springs can be created (And destroyed) at any time with a few custom properties such as size and position. Rigid Bodies have velocities, mass, collision detection and resolution, surface static and dynamic friction, drag, elasticity, and implementable collision/trigger events. The user can also navigate the scene with their mouse which allows for zooming out and changing the viewed world position.

The user is also provided a “config.txt” file which allows certain engine values or calculation techniques to be changed, as well as providing togglable debug or unstable features (e.g., debug lines and spatial partitioning). The config can also be encrypted for anyone creating a release build with the engine.

## Engine Controls

### Create

All shapes can have their sizes changed by dragging the mouse while holding their respective letter

#### Dynamic

* Create Box: Hold [S] + Drag Left Mouse Button
* Create Circle: Hold [C] + Drag Left Mouse Button
* Create Plane: Hold [Q] + Drag Left Mouse Button
* Toggle Direction of Plane Normal: [Tab]

#### Test

* Spawn constant circles: Hold [L]

#### Ropes

Ropes have no drag functionality

* Create Rope with static base: [X]
* Create Rope with non-static base: [Z]
* Increase Rope Length: [Up Arrow]
* Decrease Rope Length: [Down Arrow]

#### Kinematic

* Create Box: Hold [Left Control] + Hold [S] + Drag Left Mouse Button
* Create Circle: Hold [Left Control] + Hold [C] + Drag Left Mouse Button
* Create Plane: Hold [Left Control] + Hold [Q] + Drag Left Mouse Button

### Object selection

* Select Object: Left Mouse Click
* Delete Object: [Del] or [Backspace] when selected

### Erase/Delete

Works just like creating a shape, but while holding Left Shift at the same time. It will delete all objects that collide with the newly created shape.

* Erase as Box: [Left Shift] + [S] + Drag Mouse
* Erase as Circle: [Left Shift] + [C] + Drag Mouse

### Scene

* Move Camera: Hold [Middle Mouse Button] + Drag Mouse
* Camera Zoom: Scroll Wheel
* Camera Zoom Unscaled: [Ctrl] + Scroll Wheel

## Possible Improvements

One of the biggest improvements would be a fully implemented spatial partitioning system. Although the engine currently does have the ability to use fixed grid space partitioning, it is quite unstable and only improves efficiency in specific cases. Using a technique like Quadtrees or Octrees could improve the performance of the engine. It would also be good to support non-primitive shapes and collision detection and resolution. Allowing for polygon shapes would greatly expand how much can be done with the engine.

# Graphical user interface Description automatically generatedClass Diagram

# Third-Party Libraries

AIE Bootstrap

# Sources

Circle-Circle Collision maths: <https://ericleong.me/research/circle-circle/>

Physics and Render separation: <https://gafferongames.com/post/fix_your_timestep/>

Contact friction: <https://gamedevelopment.tutsplus.com/tutorials/how-to-create-a-custom-2d-physics-engine-friction-scene-and-jump-table--gamedev-7756>

Spatial Partitioning: <https://www.youtube.com/watch?v=7HY_SqqaoL4&t=308s>