

## Introduction

Welcome to *Golems*! This is a most unusual game. There are no set objectives, and no way to “win”. The game is a 3-D Physics Sandbox, and you are free to build anything that you would like, and watch it move realistically according to Newton’s laws.

## Structure

*Golems* is broken into two parts. The first is the **Editor**. This is the stage that you are presented when the game loads. The editor allows you to build, paint, and program your 3-D machines. The second is the **Simulator**. When the simulator is activated, the machines move naturally as dictated by the laws of physics. The simulator allows you to control the speed of the simulation, and view the action from the machines’ perspective through the use of cameras.

Activating or deactivating the simulator is done by pressing the button at the top-right corner of the screen.

## Using the Editor: Basics

The editor is very easy and intuitive to control. There are four basic skills to master: Creating objects, manipulating objects, camera controls, and using ‘wire mode’, which allows you to program your creation.

### 1. Creating Objects

To create an object (such as a cube or a sphere), simply move the mouse to the left of the screen. A menu will appear, called the *left menu*, with a list of different options. The buttons are colour-coded. White buttons allow you to manipulate objects; blue buttons create objects of different shapes; red buttons create joints; green buttons create various sensors; yellow buttons are for miscellaneous components. Try using the **blue buttons** to make basic objects: cubes, spheres, cylinders, pyramids, capsules, cones, and ellipsoids. They will appear in the centre of the editor.

Use the **red buttons** to make moving parts, called *joints*. *Golems* has four types of joints: Axles, hinges, hydraulics, and ball-and-socket joints. Axles allow for continuous rotational motion in one direction, like a car wheel. Hinges restrict rotational motion to 180 degrees, like on a door. Hydraulics allow for linear motion – moving forward and backward. Ball-and-socket joints allow for rotation in all directions.

## 2. Manipulating Objects

### a. Moving Objects

To move an object around in the editor, **left-click on the object and drag the mouse**. When the object is selected, it will turn bright green, and a *movement direction disk* will appear. This indicates the plane of movement of the object. In three-dimensional space, there are three planes of movement: The XZ-plane, which is horizontal; the XY-plane, which is vertical; and the YZ-plane, which is also vertical. The XZ plane is the default plane of movement. Holding ‘z’ or ‘x’ on the keyboard will change the plane of movement, and the movement direction disk will follow accordingly.

To make precise movements, use *grid mode* or *axis mode*. Grid mode, activated with the ‘**caps lock**’ key, creates an invisible grid of points are moved on. This makes it very easy to line up a wheel with an axle, for example. Axis mode is activated with the ‘**tab**’ key, and restricts movements to one direction at a time.

### b. Rotating Objects

Objects can also be rotated. To rotate an object, **hold the shift key and left click on the object, then drag the mouse**. The shift key changes the movement direction disk to the *rotation direction disk*. This indicates the axis about which the object is being rotated. Again, in 3D space there are three axes of rotation: X, Y, and Z. The Y axis (vertical) is the default axis of rotation. Holding ‘z’ or ‘x’ on the keyboard while holding shift will change the axis of rotation, and the rotation direction disk will follow accordingly. An alternative to using the shift key is to use the white *rotation button*, on the left menu.

To make precise rotations, use *grid mode*. By default this restricts rotations to increments of 15 degrees.

### c. Stretching Objects

To stretch an object, select the object with **left-click**, and press ‘**alt**’. Alternatively, press the white *resize button* on the left menu and select an object. A series of *control points* will appear which are specific to the type of object. Control points are white spheres that can be moved back and forth to control the dimensions of an object. For example, cubes have six control points, allowing each face to be stretched independently. A sphere has only one control point, which sets its radius. **Left-click on the control point and drag** it with the mouse to alter the size of an object.

To make precise stretches, use *grid mode*. This locks size increments to the grid.

### d. Copy, Delete, Undo, Redo

To copy an object, select the object (with **left-click**), and press ‘**c**’. This creates an identical object at the same location as the original. To delete an object, select the object and press the ‘**delete**’ key. You can undo a mistake at any time by pressing **control-z**, and redo it by pressing **control-y**.

#### e. Selecting Multiple Objects

There are two ways of selecting multiple objects. If the objects are in contact with each other (an arrangement called a *group*), you can use *group-select mode*. On the left menu, press the *group-select mode button*. Clicking on any member of the group will select the entire group. To return to moving one object at a time, press *single-select mode* in the left menu.

If the objects are not in contact with each other, you can use *multi-select mode*. Hold the '**control**' key and click on several objects to select them one at a time. Note that you can combine group-select mode and multi-select mode to select several groups.

### 3. Camera Controls

The camera is manipulated with the mouse. To pan the camera, **left-click and drag** the mouse about the screen. You will see the camera follow your mouse. To zoom the camera, **roll the scroll wheel**. Rolling forward zooms in, and backward zooms out. To rotate the camera, **right-click and drag** the mouse about the screen. To focus on an object, select it and press '**f**' (for *focus*). If focus is pressed with no object selected, the camera returns to its default orientation.

### 4. Wire Mode

*Wire mode* allows you to make your machines intelligent, by powering joints and reading sensors. To access wire mode, press the *Wire mode button* in the left menu. When in wire mode, all physical objects will appear in a transparent blue colour. Joints will appear with brightly-coloured *terminals* in two colours: yellow (output) and green (input).

#### a. Battery Basics

The green *battery button* in the left menu creates a *battery*. This is a source of power for joints. Creating a battery takes you automatically to wire mode. The battery has one yellow output terminal. Batteries can be moved, rotated, copied, and deleted like any other object.

To power a joint, simply **left-click** on the output terminal of a battery and **drag** to the input terminal of the joint. Doing so creates a *wire*, in yellow, which indicates that there is a connection. The wire carries the signal from the battery to the joint, causing it to actively spin (for an axle or hinge) or push (for a hydraulic). A powered joint is referred to as a *motor*.

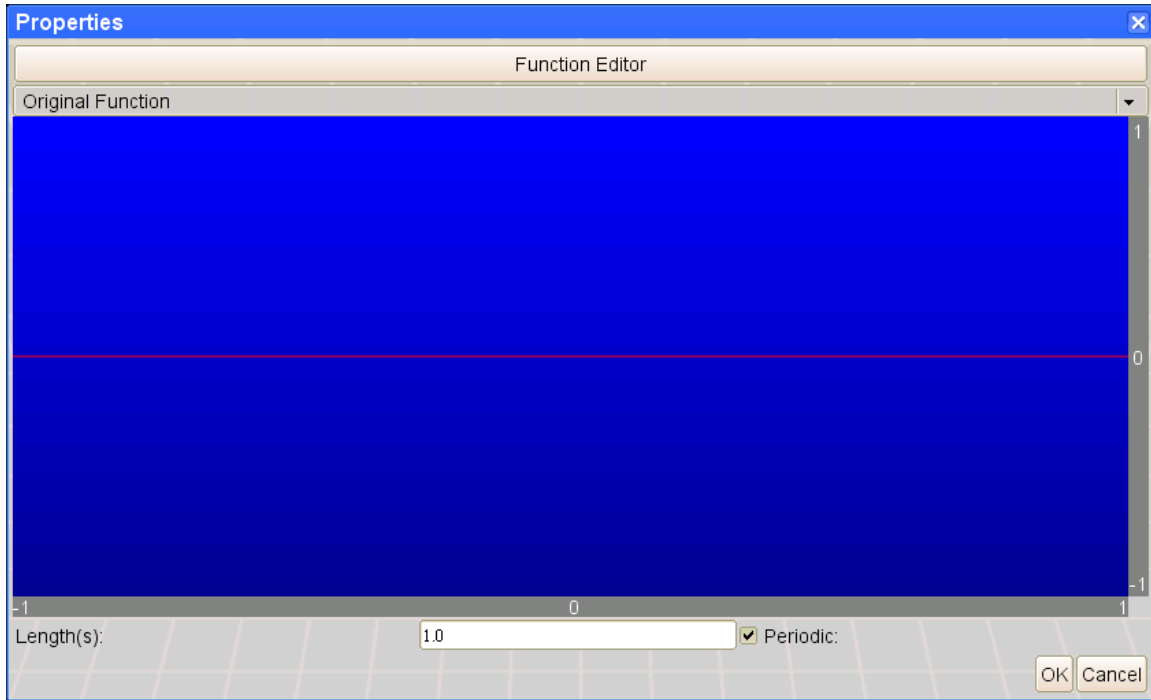
#### b. Wire Basics

Wires carry a signal from the output of one object to the input of another. They are normally *yellow*. Sometimes it is convenient to be able to change the polarity of a signal - for example, if the motor is spinning the wrong way when powered. The easiest way to do this is to use a *red wire*. Hold the '**alt**' key when creating a wire, and that wire will turn red. A red wire carries the opposite-polarity signal of a yellow (positive) wire. If a wire is created between two terminals that are already connected, the new wire will replace the old one.

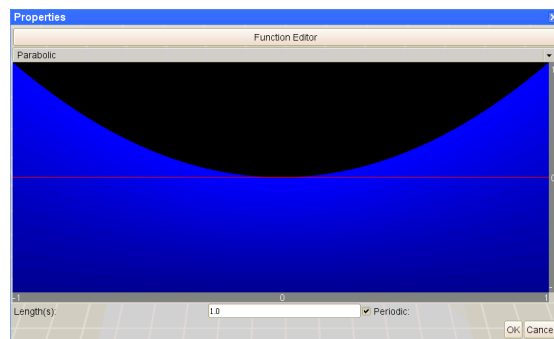
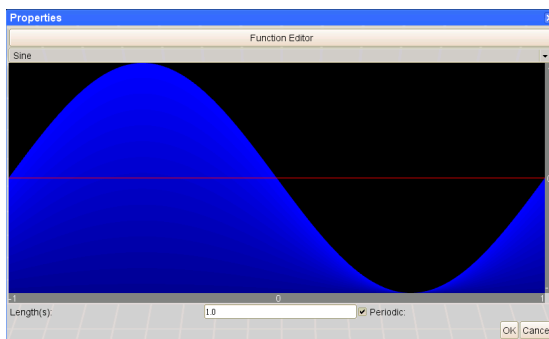
Wires can also be deleted. To delete a wire, select it with **left-click** and press the ‘**delete**’ key, just like any other object.

c. Functions

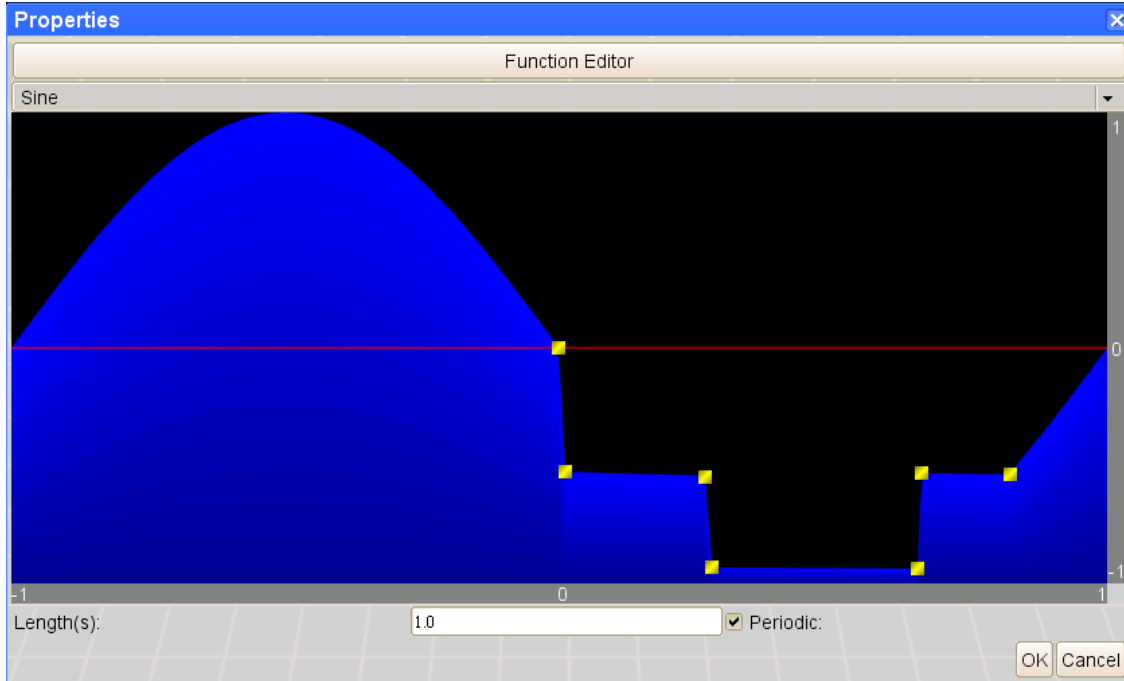
An important part of wire mode is the *function editor*. Normally when a battery is created, it sends a steady, maximum-power signal. However, it’s easy to make it send signals that vary over time. To do this, select the battery and press the ‘**enter**’ key. This opens the function editor.



The function editor consists of a window that displays a graph of the signal over time. The y-axis represents the *amplitude* of the signal. This is always a number between zero and one. The x-axis represents the *time*, in seconds. The total time duration represented on the graph can be varied using the *Length (s)* box. If the function is made *periodic*, then it will repeat from the beginning after this time has elapsed. You can choose from different pre-made functions using the drop-down menu:

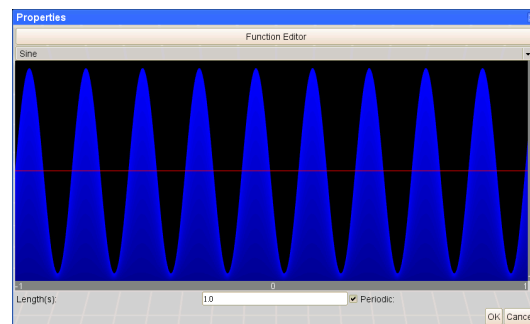
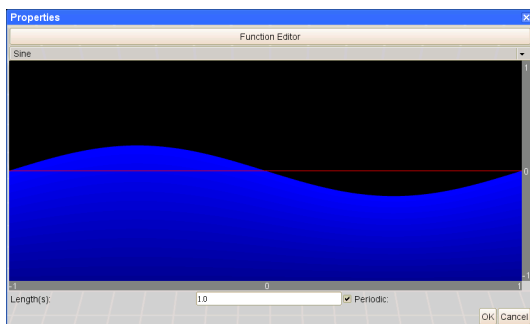


Or create your own:



To create your own functions or modify the given ones, simply **left-click** within the window to add points. These points can be dragged to new locations by selecting them with **left-click** and dragging the mouse, or removed by selecting them and pressing the **'delete'** key.

The entire function can be rescaled by moving the mouse to the *y-axis bar* on the right and using **right-click and drag**. Likewise, the time axis can be rescaled by moving the mouse to the *time-axis bar* and again using right-click and drag. The function can be shifted by moving to an axis bar and using **left-click and drag**. This translates the entire graph up, down, left, or right.



The example on the left would apply a very weak signal that varies slowly, while the example on the right would apply a very strong signal that varies quickly.

# Buttons Dictionary

These are all of the buttons in the left menu, and a summary of their purpose.

## 1. White Buttons



### Single Select Mode

In this mode, only one object at a time will be selected.



### Group Select Mode

In this mode, any objects in contact with each other will be selected.



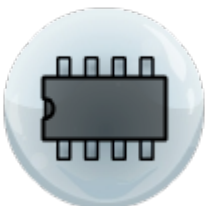
### Resize

This creates control points for stretching an object without pressing **alt**.



### Rotate

This enables rotation of objects without pressing **shift**.



### Wire Mode

This changes the view to wire mode. Objects become transparent blue, and while wires and functional components become visible.

## 2. Blue Buttons



### Cube / Rectangular Prism

Creates a cube. The cube can be stretched into any rectangular prism using its six control points.



### Sphere

Creates a sphere. The sphere has only one control point, which sets its radius.



### Cylinder

Creates a cylinder. The cylinder has three control points, for determining its radius and height.



### Pyramid

Creates a pyramid. The pyramid has three control points – one for height, and two for its base dimensions.



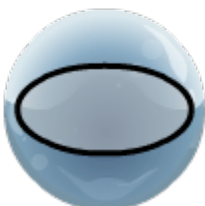
### Cone

Creates a cone. The cone has two control points – one for height, and one for radius. **Note that cones are not efficient for physics simulation, so avoid use unless necessary.**



### Capsule

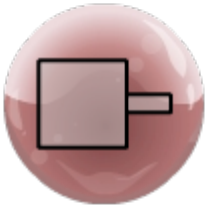
Creates a capsule – a cylinder capped with hemispheres. The capsule has three control points, for determining radius and height. Note that capsules are very efficient for physics simulation.



### Ellipsoid

Creates an ellipsoid – a ‘stretched sphere’. The ellipsoid has three control points, for stretching each axis. **Note that ellipsoids are very inefficient for physics simulation. Avoid use when possible.**

### 3. Red Buttons



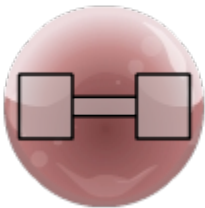
#### 'Axle' Rotation Joint

This joint allows complete continuous rotation along one axis. When powered, it will spin. It can be set to output its angle, its rotational velocity, or its rotational acceleration.



#### 'Hinge' Rotation Joint

This joint allows pivoting rotation along one axis. The pivot can swing at most 180 degrees. The starting angle of a hinge can be set by using the resize control points. Hinges are very efficient, and can be linked into large chains, or looped to form 'tank tracks'.



#### 'Hydraulic' Linear Joint

This joint allows for linear motion along one axis. When powered, it will extend if it receives a positive signal, or contract if it receives a negative signal. The linking bar of the hydraulic is not a physics object, and can therefore pass through other objects.



#### 'Ball and Socket' Rotation Joint

This joint allows for freedom of rotational motion in all directions. However, it cannot be powered. In the current release version (0.1), ball and socket joints cannot be resized.



#### 4. Green Buttons



##### Battery

The battery outputs a programmable, time-varying signal.



##### Unary Math

The math operator has a programmable function just like the battery. Instead of a time axis, the 'x' axis is the *input signal*. The input gets mapped to a new output signal. The math operator also can act as a 'differentiator', outputting the rate of change of a signal.



##### Touch Sensor

The touch sensor reports collisions with nearby objects. It outputs a '1' when a collision occurs, and a '0' at all other times.



##### Radar Sensor

The radar sensor detects far away objects. It comes with a "sensor field" that can be stretched into various shapes. When the sensor field touches another object, the radar sensor outputs a '1'. At all other times it outputs a '0'.



##### Multimeter

The multimeter is a general-purpose position sensor / orientation sensor / speedometer / accelerometer. It outputs a signal between 0 and 1, depending on the measurement range set in its preference menu. For certain settings, it has several outputs – one per axis.

## Yellow Buttons



### Rocket

When powered, the rocket exerts a force to propel itself forward. The maximum force can be set in the rocket's properties menu.



### Camera

Cameras are added in wire mode. The '**Page up**' and '**Page down**' keys cycle through the cameras that have been added, giving their perspective. When connected to an object, the camera will move with it, allowing you to get the object's viewpoint.

## Simulation

Running the simulation is as simple as pressing the *run* button at the top-right corner of the screen. As soon as this is done, objects will fall to the ground according to gravity, motors will spin, and collisions will occur, all according to Newton's laws of motion. The speed at which the simulation runs defaults to real time, but can be changed using the main menu (which is accessible at any time by pressing the '**escape**' key).

### 1. Physics Engine

*Golems* uses the Open Dynamics Engine (<http://www.ode.org/>), an open-source rigid-body simulator. ODE is a fast, stable, and efficient simulator, but not perfectly accurate. For example, it is not guaranteed that running the same simulation twice will give the same outcome – there is a bit of randomness involved.

### 2. Efficient vs. Inefficient Structures

What machines you can run is generally limited only by the speed of your computer, but there are some important points to remember. ODE does not have efficiently implemented ellipsoids or cones; these are approximated by trimeshes and a series of cylinders, respectively. These shapes are much more computationally more expensive than cubes, spheres, and capsules, which are the most efficiently implemented shapes.

## Notice

This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.