Billiards Manual

Billiards is a free cue sports simulator. It aims for physical accuracy and simplicity and should hopefully be useful for practicing billiards on your own and against your friends when a real pool table is not available. It is also designed to be customizable enough to allow one to tailor the simulator to his needs (to implement new games for instance or to study the physics of the game by fiddling with the physical properties of the equipment, setting up shots or plotting ball trajectories, velocities and spin).

This manual describes how to use and customize Billiards.

Table of Contents

1 Overview

Billiards is a physical simulator. Its basic function is to draw a table, a set of balls and a cue stick and to calculate the physical interaction between them as accurately as possible. That happens to be its only function as well. It does not keep score nor does it attempt to enforce any rules and such is the case for two reasons: First of all my own interest lies in simulation, animation and drawing, therefore my main concern is not Billiards itself but instead Techne, the platform Billiards is running on. After spending some time to implement and debug such features up until version 0.1 I got bored and gave up. I think Billiards is actually more useful this way but not anyone may share this opinion and this brings us to the second reason. As previously mentioned Billiards is not a standalone application but is actually layered on top of Techne, a programmable simulator and renderer and this makes it particularly easy to customize or extend Billiards. You can see some examples of such extensions in the chapter on Chapter 3 [Hacking at the internals], page 5. Should you therefore find that there are some important features that Billiards is missing, you're no longer forced to stay idle, in hope that the author will at some point add them. (Such hope probably doesn't exist in reality anyway). Implementing them yourself is probably easier than you think and, if not, my email address is always available in case assistance is required.

Moving away from what Billiards can *not* do lets discuss what it *can* do. I'm under the impression that Billiards' physical model is at least to some extent accurate. I've tried to use whatever publication I could get my hands on in order to finetune the various parameters that come into play, mainly the coefficients of friction and restitution for collisions and contact between the balls, stick and cloth. All experiments carried out in order to test the validity of the model seemed to yield realistic results but what gives me more confidence is the fact that I didn't have to add any features 'by force'. Let's consider cue ball squirt for example. In its early stages of development Billiards would exhibit the following behaviour: When english was applied while striking the cue ball, it would start moving along a deflected path almost as if you had miscued. I initially tried to remedy this by artificially setting the direction of the force applied by the stick to the cue ball to be that of the cue stick deflected by a small squirt angle which I had calculated theoretically. That worked but it did not address the root of the problem which lied in the way I had modeled the stick-ball collision. When a player holds a cue stick, it is mainly allowed to move in the direction parallel to its axis but it can also deflect sidewards to some extent, depending on the bridge type employed by the player. It is easy to show that this freedom to deflect is crucial in keeping the ball in its intended path but this is not the point. The point is that once this was taken into account everything was back to normal. Not only did the ball move along the expected path but it also exhibited squirt and, once a few parameters such as the sticks inertia and the stick-ball friction coefficient were determined through experiment, the squirt angle was found to agree pretty well with experimental measurements. What's more, the player would now find that she could not apply infinite english any more, not without miscueing at least and the miscue limit would also agree with theory once correct parameters were determined.

There are a few areas where the model is lacking nevertheless. The first and most prominent should be jumpballs. The problem is that they're not at all possible. The cue ball or object balls can be driven off the table easily enough if you apply enough english

and force but you can't jump the cue ball with the cue stick alone by elevating the cue and using bottom english. This is due to the fact that all objects in Billiards are considered completely rigid I believe. That is, the elasticity of the cue tip and playing surface should be taken into account to get jump shots to work. Elasticity may play an important role in other cases like ball-cushion collisions as well. In this area the model employed is quite simplistic and therefore probably inaccurate. I'm not sure if ODE has enough support for flexible bodies to correctly implement such effects. This needs to be further looked into.

You can use Billiards' physical model to play pool and carom games. Snooker isn't supported as I haven't worked up the courage to put together a suitable table for it yet. Apart from that Billiards also sports some nifty graphics if I may say so myself, but only if your GPU and, for the time being at least, your drivers can support the needed features. If this is not the case there is a plain toon-style renderer which should run on all hardware. There's also a rather eccentrically implemented GUI meant to interfere as little as possible with your playing and which you can access through Techne's browser or, alternatively your favorite web browser. That's more or less it, no sound or networking for now although I hope this will change in the future. Before moving on let me say that I would be very interested in feedback from more experienced players. Feel free to contact me by e-mail if you have any comments or suggestions, especially regarding the physical model.

2 Running and playing Billiards

This chapter describes how to play pool or carom in Billiards assuming you already know enough to play them on an actual table. If this is not the case there are plenty of resources on the web that you can use to learn the rules of the various games so I won't try to explain them here. I shoud say nevertheless that it would be very useful if we could have a few chapters dedicated to how you can use Billiards to learn each game. It would be very easy, technically speaking, to set up shots in Billiards and create courses where you can practice the skills and disciplines needed for each game but I'm lacking both the theoretical knowledge and the practical skill needed to create such courses. I expect Billiards to be of use to more experienced players though, so if you fit that description and want to take up the job let met know. Think about it, you'll get to be a coauthor in a book. Wouldn't that be grand?

Assuming Billiards and Techne have both been installed properly on your system you can run Billiards simply by typing `billiards-browser`[1] on the command prompt or possibly though the window manager's menus if you have installed a packaged version that supports this. During start-up, the file '`.billiards`' in the user's home directory is read and executed. The purpose of this file is to hold per-user configuration but bear in mind that it is a normal Lua script. This allows more flexibility compared to the usual option-value pair syntax, should it prove useful. For more information on how to edit this file to your taste see Chapter 3 [Hacking at the internals], page 5.

The only part of the simulation that cannot be configured via this script is the low-level video and audio configuration. The bit depth of the red, green, blue, alpha, and depth components of the frame buffer as well as the sampling frequency and refresh intervals of the audio device can be specified on the command line only. You can run Billiards with the '`-h`' or '`--help`' switch for the relevant command line arguments although it's rather unlikely that you'll ever need to change these. They mostly exist for the sake of the poor bastards who have to debug Billiards and this currently only includes myself.

This probably leaves '`-O`' or '`--option`' as the only useful option for most users. With this option you can override configuration values so that you won't have to edit the configuration file every time you want to change certain common parameters such as the type of table you want to play on. You can currently specify the following options:

'`eightball, nineball and carom`'
: Start an eightball, nineball or carom game straight away without waiting for the web interface. A useful shortcut if your familiar with the command line.

'`toon`'
: Use the simpler cel shader. This should run on all hardware.

'`notoon`'
: If Billiards decides to use the toon shader by default, because it doesn't think much of your GPU, but you want to prove it wrong you can try this option. You probably won't have much luck but it's worth a try.

'`framerate`'
: If you need confirmation that your computer is indeed faster than your friend's specify this option to get a framerate counter.

[1] You can also start billiards with the command `billiards` instead. This will not start Techne's browser so you'll have to use another browser to visit `http://localhost:29176` in order to start a game unless you specify the game type as an option on the command line.

‘`nohttpd`’ Don’t start the web server.

‘`annotate`’
This annotates each ball with its current speed an spin when highlighted. Mainly useful for debugging purposes.

‘`slow`’ If this option is used simulation goes into slow motion as soon as the cue ball is struck. Can be useful when practicing, experimenting or debugging.

As an example, to start a nineball game in normal shading you would have to run Billiards like this:

```
billiards -Onineball
```

You may also have to specify the option ‘`notoon`’ if toon shading is the default so in that case you’d issue the command:

```
billiards -Onineball -Onotoon
```

Once you’ve started it, playing Billiards is simple enough. Initially you’re in looking mode. Hold down the left mouse button and move the mouse to look around or use the right button to zoom in and out. You can get a panoramic view of the table by holding the middle mouse button. You can also move the balls around when looking in order to cheat or set up a shot. Just highlight the ball you want to move with the mouse pointer and drag it around with the right mouse button. When you’ve decided how to perform the shot, highlight the cue ball and click on it to enter aiming mode. You can also select any other ball while looking as the cue ball by clicking on it. In aiming mode the cue stick appears and you can use the left mouse button to fine-tune your aiming. Highlighting the cue tip and dragging with the left mouse button controls english. To elevate the cue you need to highlight the cue butt and drag with the right button. When you’re happy with your aim drag the butt with the left mouse button to stroke the cue and perform the shot. If you change your mind click on the cue ball again to enter looking mode and reconsider the shot. Once the cue ball is on its way you can look around and zoom in the usual way while waiting for the balls to come to rest. Apart from that you can use your mouse wheel when no ball is highlighted to hasten or slow down the flow of time and that’s all you need to know to play the game.

All the buttons mentioned are the default bindings which can be changed if you don’t find them convenient. See Section 3.5 [Bindings], page 9 for details.

Finally note that it is very easy to strike the cue ball too hard in Billiards. Bear that in mind if the effect of english (particularly follow or draw) seem exaggerated or if you keep knocking balls of the table.

3 Hacking at the internals

The various options described in the section on running Billiards (see Chapter 2 [Running and playing Billiards], page 3) simply configure certain common parameters of the game for you. There is much more you can change but in order to accomplish that you have to learn how to edit the configuration script. Fear not though, read on.

3.1 Introduction

The first thing you must do in order to set parameters is to locate and open the configuration script with your favorite editor. This script is a file named '`.billiards`' and it should reside in your home directory. If it does not exist[1], or if you have deleted it, simply run Billiards once and it will be created with default values.

The parameters available for you to tamper with are divided into three groups. In order to set the parameter 'bar' within group 'foo' you have to add a line of the form `foo.bar = value` to the configuration script (or locate a possibly existing line and change the value). The value is usually a number but it can also be a set of numbers enclosed in braces, a string enclosed in single or double quotes or a boolean value (either 'true' or 'false'). If all this doesn't make any sense to you, taking a look at the configuration script might help. Changing the default values found there is very straightforward. Here are a few lines:

```
dynamics.stepsize = 0.0012
dynamics.iterations = 0
dynamics.gravity = {0, 0, -9.81}

billiards.tablewidth = 2.84
billiards.tableheight = 1.42
billiards.cushionheight = options.pool and 0.033 or 0.037
```

3.2 Game options

The following tables describe the available parameters in each category. We begin with the 'billiards' category which contains the configuration parameters of the game itself and of the equipment used.

'`tablewidth, tableheight`'
: The playing area of the table in meters. The standard size for a tournament sanctioned by the UMB is 2.84 by 1.42 meters. This parameter should not be changed for now. It won't work until Lua 5.2 is released and Techne adopts it.

'`cushionheight`'
: The height of the cushions in meters at the point where they contact the balls.

'`ballradius`'
: The radius of the balls in meters.

'`ballmass`'
: The weight of the balls in kilograms.

[1] Beware the files that start with a dot are considered as hidden.

'`cuemass`' The mass of the cue in kilograms. Choose this according to the weight of your favorite cue.

'`cueinertia`'
This is the effective moment of inertia of the cue tip around the vertical axis. This value affects the initial deflection of the path of the cue ball due to left or right english, also known as the squirt effect. Make the value smaller to get that low squirt cue you always wanted or fiddle with it to match your own cue.

'`cueforce`'
The maximum force the player can exert while stroking the cue in Newtons. This value is not very useful although making it smaller might make it easier to perform gentle shots.

'`cuelength`'
The length of the cue in meters. There's no need I can think of to change this value. Leave it be.

'`staticfriction, slidingfriction`'
The coefficients of static and sliding friction between the cloth and balls. This determines how soon the ball begins to roll without slipping after it has been struck by the cue. Typical values for sliding friction are around 0.2 for modern tables with larger values being characteristic of tables manufactured in the past. Static friction has been empirically found to be a little smaller, about 0.14 which is the value used by default.

'`rollingfriction`'
The coefficient of rolling friction between the cloth and balls. The lower the value the longer do the balls roll before coming to a stop. Divide 1 by this number to get the 'table speed'. The default value is 0.01 which yields a table speed of 100.

'`spinningfriction`'
The coefficient of spinning friction between the cloth and balls. This determines the amount of friction present when the ball spins like a top around the vertical axis. Larger values therefore tend to make the balls lose the vertical component of their rotation faster.

'`slowfriction, fastfriction`'
The coefficient of sliding friction between two colliding balls. This value is normally highly dependent on the relative collision velocity with fast collisions typically yielding much smaller coefficients of friction. You must therefore specify the coefficients for both slow and fast collisions. Higher coefficients of friction result in better spin transfer between balls and thus more exaggerated throw effects. Lower values correspond to smooth, well polished balls.[2]

'`bouncingfriction`'
The coefficient of sliding friction between the balls and the cushions.

[2] In case you need to know, the actual formula used to compute the coefficient is $b+(b-a)\exp(-0.77*V)$, a simple exponential interpolation.

‘`strikingfriction`’
: The coefficient of friction between the cue stick and the cue ball. This value determines how well the cue tip grips the cue ball and therefore how easy or difficult it is for you to miscue. Large values correspond to a well chalked cue while smaller ones can be used to simulate a bad or worn cue.

‘`collidingrestitution`’
: The coefficient of restitution between colliding balls. This defines how efficiently the balls collide with each other and should usually be very close to 1.

‘`bouncingrestitution`’
: The coefficient of restitution between balls and cushions. Higher values result in more elastic and thus efficient cushions. Typical values fall around 0.7.

‘`jumpingrestitution`’
: The coefficient of restitution describing the elasticity of the table bed. Large values make it easier for a ball to bounce off the table.

‘`strikingrestitution`’
: The coefficient of restitution between the cue tip and the cue ball. Choosing a larger value should result in a more elastic cue (although I doubt you’ll see a difference).

‘`thresholds`’
: A set of two values with the linear and angular velocity in meters and radians per second respectively that a ball must have in order to be considered at rest. Changing this can be useful if a shot never finishes although all balls seem to be resting.

‘`tracking`’
: A set of two thresholds specifying the resolution of the trajectory curves displayed in the shot history section of the browser. If you don’t understand this you probably don’t need to change it.

‘`linear`’ This value affects the sensitivity of the camera armature while the camera is zooming in or out.

‘`angular`’ Similar to ‘`linear`’ described above but affecting camera rotation.

‘`finetune`’
: Similar to ‘`angular`’ above but affecting camera rotation when finetuning in aiming mode.

‘`stroke`’ Change this to fine-tune the stroking of the cue with the mouse.

3.3 Graphics options

The ‘graphics’ category contains global options concerning how the table and balls are drawn, mainly resolution, field of view and the like.

‘`window`’ The size of the graphics window in pixels. Setting this to ‘nil’ or ‘false’ hides the window altogether.

‘`title`’ The window title string.

'`canvas`' A RGB color triplet defining the color of the background.

'`perspective, orthographic`'
These are sets of values describing the 'projection transformation'. The main reason you might care to change it would be to set the field of view which you can accomplish easier by using the 'derived' configuration table.

3.4 Simulation options

The 'dynamics' group affects the way the simulation is performed. The main option to fiddle with is the '`stepsize`' which affects the accuracy of the simulation. Set this as low as your CPU allows.

'`timescale`'
Think of this as the fast-forward button. Setting this to values greater than 1 will make time pass faster. Smaller values will result in slow motion.

'`stepsize`'
The simulator stepsize. The rule is: the smaller, the better, the slower.

'`iterations`'
Setting this to a value other than zero will switch to a different simulation method which is faster but also less accurate. More iterations make this method more accurate (but still less accurate than zero iterations) and also slower (but still faster than zero iterations I believe).

'`gravity`' The gravity vector. Set this to something other than `{0, 0, -9.81}` only if you want to experience shooting pool in space.

'`surfacelayer`'
You can probably safely ignore this unless you understand what it does and have reason to believe it can help you resolve some problem you're having. This option defines the depth at which bodies may sink into a surface before coming to rest. This ensures that precision errors are avoided which might cause bodies to repeatedly establish and break contact when they should be resting on the surface.

'`popvelocity`'
This defines the maximum velocity the simulator can give to a body in order to push it out of some surface it has penetrated. You might want to try to lower this if you see balls jumping around for no apparent reason but you can probably ignore this as well.

'`tolerance`'
A set of two values, the first describes how much the simulator will allow a joint (such as a contact between two bodies) to be violated. Setting this as low as possible is beneficial to accuracy but may lead to instabilities and weird behavior. The second value describes how much of the accumulated joint error the simulator tries to correct at a time. Setting this to higher values has the same effects as described above.

'`ceiling`' This is a time threshold. If a frame takes more than this number of seconds then no simulation is performed during that frame. If we did simulate the next

frame would probably take even more and so on. This would make Billiards appear frozen for all practical purposes.

Finally there's a 'network' group. The only useful option in this group so far is:

'`http`'
: The port the HTTP server will listen on for requests. If you choose, say 12345 then you can find the pages at `http://127.0.0.1:12345/`.

There's also one more table which is not a real configuration table like those previously described. This table allows you to set some of the same values as those described above, but in a simpler way. It is therefore called the 'derived' table

'`width, height`'
: The width and height of the window in case you need to update them separately.

'`softness, stiffness`'
: Again the values of the 'tolerance' option in the 'dynamics' table separately.

'`field`'
: The field of view in degrees. Values around 40 should work for most people. Try larger ones if you're curious as to how fish would play billiards if they had fingers.

'`gee`'
: The acceleration of gravity as in the 'gravity' option in the 'dynamics' table but this takes a single value and assumes that the vector points downwards as is to be expected in most situations.

3.5 Bindings

Another set of useful options are the bindings, that is the specific mouse buttons that are associated with game functions like rotating, zooming or quitting. All bindings reside in the `bindings` group and should be given numeric values, 1 for the first or left mouse button, 2 for the second button which is usually the wheel and so on. As an example here are a few lines taken from the default binding configuration:

```
bindings.ready = 1
bindings.survey = 2
bindings.rotate = 1
bindings.move = 3
bindings.pan = 2
bindings.zoom = 3
```

The currently defined bindings are:

'`rotate, zoom`'
: These are the buttons used to rotate or zoom the camera and are initially bound to the left and right mouse button respectively.

'`survey`'
: Change the camera to an overhead view of the whole playing area. Default is the second (middle) mouse button.

'`ready`'

This is the button with which you have to click on the cue ball in order to toggle in and out of aiming mode. Normally this is bound to the left mouse button.

'`elevate`'

This enables cue elevation when the cue butt is highlighted and is by default bound to the right mouse button.

'`strike`'

The mouse button used to stroke the cue when the cue butt is highlighted. Normally the left mouse button.

'`offset`'

This is the button to hold down when the cue tip is highlighted to apply english to the shot. Normally it's bound to the left mouse button.

3.6 Hooks

Sometimes it might be desirable to be able to change options dynamically, at certain moments during the course of the game. For example you might want to study a particular shot in slow motion. The best way to do this would be like this:

```
billiards.cuecollision.goslow = function ()
   simulator.timescale = 0.25
end
```

This defines a 'hook', that is a function that will be executed as soon as the event associated with that particular hook takes place. As already mentioned the whole configuration file itself is indeed also a function and therefore you can think of hooks as mini 'configuration blocks' that will take effect upon a specific event. In this case the simulator is set to run 4 times slower than usual but only after you have stricken the cue ball.

If you need more than that, you should probably learn the language that all these scripts are written for, namely Lua. Lua is simple and easy to learn and at the same time very efficient, particularly for describing and processing structured data. It is by no means a simplistic toy language or anything of the kind. You can do everything you can do with most general purpose scripting languages without having to learn tons of syntax rules.

Most hooks of interest reside in the 'billiards' group. These include:

'`looking, aiming, waiting, striking`'

These hooks fire at the various phases of a shot: when the shot begins and you look around the table, when you start aiming, and so on. Note that the shot might be interrupted at any phase and a new shot started, so don't rely on the natural order of phases. Always try to clean up in your `looking` hook.

'`ballcollision, cushioncollision, cuecollision`'

Use these hooks to have something happen each time the balls collide with each other, with the cushions or with the cue. Don't expect these hooks to be called

only once for each collision though. They'll be called during each timestep, as long as the colliding surfaces are still in contact.

'`reorienting, adjusting`'
: These hooks are executed when you rotate or zoom the camera (in the case of the '`reorienting`' hook), or when you adjust cue elevation and english (for '`adjusting`'.

'`grabbing, releasing`'
: Called when you grab and start moving around a ball or when you put it down again.

There are also some hooks in the 'graphics' group which might be of interest although they're not part of Billiards but of Techne itself. These are:

'`focus, defocus`'
: These hooks fire when the window Billiards is being displayed in is focused and unfocused.

'`close`' Techne calls this when you close the window by clicking on the close button in its title bar.

Finally, there is a hook named '`collision`' in the 'dynamics' group but you're advised against fiddling with this hook. It fires many thousands of times per second (each time a collision is detected) and can therefore slow things down considerably if used carelessly.

3.7 Hacking deeper

What has been described so far is not the only thing you can do in order to fiddle with Billiards, it's just a set of preprogrammed options designed to be useful and easy to change. You can, in fact, include arbitrary Lua code in your '`.billiards`' file and change every aspect of the game or implement a different game altogether. Billiards is, after all, nothing more than a set of files such as this one together with a set of resources (textures, geometry and the like) which are again Lua scripts. You might need this to implement a different initial ball setup for snooker say but in order to know *what* you can do and *how* you can do it you need to do some reading first. Start with the Lua manual or, if you prefer it, an introductory tutorial on the Lua language and continue with Techne's manual, at least some introductory material to get an idea of how Techne works. This last manual has not been written yet but I hope to cook up a draft at least when I'm done releasing Billiards, Techne and Aviation. If you've gotten so far, you're ready to start browsing through Billiards' source to see how it is implemented and, therefore, how it can be extended. All this may not sound too easy, and in fact it isn't but it's not terribly hard either. I'll be of whatever assistance I can if you're prepared to attempt it and I may also be bothered to document Billiards' internals upon popular demand. Let me know.

3.8 Performing experiments

As a general-purpose billiards simulator Billiards is very well suited for experimentation. It is relatively easy to set up a shot and execute it automatically as many times as you wish potentially varying shot parameters for each iteration. In order to make this a little easier Billiards now has a separate experimentation mode. You can access this mode with the

'`experiment`' option. This hasn't been mentioned until now but options can also take values. The value given to the '`experiment`' option is the path to a Lua script that implements the experiment. What this means is that you must start Billiards like this:

```
billiards -Oexperiment=/path/to/experiment.lua
```

Inside this script you can use any code you like to set up your experiment. In order to facilitate this experimentation mode uses a few more parameters. These are located in the 'billiards' group:

'`tipoffset`'
: This determines the distance of the cue tip from the cue ball when entering aiming mode. This is not necessary for a manual shot but when the cue is fired automatically given a target speed, it may need some space to accelerate to this speed.

'`cuesetup`'
: This is the setup of the cue when entering aiming mode. A vector value containing, in order: cue heading, cue elevation, tip sidespin and tip follow. This can be used to set up the cue as needed for the given experiment so that you don't have to do it manually for each shot.

'`cuespeed`'
: The target speed of the cue. If this is set to a number then shots are fired automatically and repeatedly by accelerating the cue to the given velocity.

'`opening`' This is a table of vectors each defining the opening position of a ball on the table. The number of position vectors in the table also implicitly defines the number of balls in the experiment. Use this in your setup hook to set up each iteration.

'`decals`' This is a table of RGB triplets (or textures) for each ball on the table. It must be specified together with opening to specify the color of each ball.

There are a couple of new hooks as well, again in the 'billiards' group:

'`setuprun:`'
: This is executed before a new run takes place and is given the number of the run as a parameter. It can be used for bookkeeping or to set up the run by setting billiards.opening to place the balls and the aforementioned parameters to set up the cue.

'`recordrun`'
: As above but executed at the end of each run. Mainly necessary for bookkeeping.

Once you've set up your experiment properly you won't have much need for visualization. Keep in mind that you pass the '`-e`' switch to Billiards to let Techne know that you're only insterested in simulation. You can then leave Billiards running in the background until its done. It will run much faster too.

3.9 The HTTP interface

It is true that the user interface in 'Billiards' is quite rudimentary. This is a result of both necessity and purpose. The system on which 'Billiards' runs is complex enough in itself leaving little time to consider implementing a full-fledged widget system on top of it. On the other hand the widget systems the user usually deals with have come a long way in terms of efficiency and appearance. I therefore find the user interfaces in most games where you can't even copy/paste your name or the IP address of your opponent very irritating. They also tend to get in the way, forcing you to move and click the mouse several times in order to start a game even though you play with the same options most of the time. Unfortunately there's no denying that they *can* be convenient.

Therefore 'Billiards' does sport a graphic user interface but the approach followed is somewhat different. The game starts with the default configuration specified in the user's '`.billiards`' file but it also sets up a small web server. After running the game you can therefore open your favorite browser[3] and connect to your own machine (the IP is '127.0.0.1') at the port you specified in the configuration file (with the default port of 29176 the URL would be `http://127.0.0.1:29176`). If you're running Billiards 0.4 you can start the web interface via Techne's own browser by running `billiards-browser` on the command line. Once the interface is up follow the links to start a new game, manage your shots or set system options. That's the beauty of using pre-existing and standard methods and tools to solve your problems: you don't have to learn a new interface (chances are you'll be used to navigating web pages and filling out forms) and I didn't have to write anything, at least not from scratch. And on top of that the HTTP interface is probably more powerful and flexible than what I would have written.

[3] Some of the pages make use of SVG graphics in order to draw diagrams of the shots etc. Make sure your browser supports SVG rendering if you want to see them.

Appendix A GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and

that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

   A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called

an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.