

cellsDS User Manual

Introduction

CellsDS is a flexible, programmable grid-based music sequencer for the Nintendo DS. In fact, cellsDS is *six* sequencers in one, with each sequencer being controlled by a user-definable script. This document covers what you need to get started. Once you get comfortable using cellsDS, you may also wish to read the Developer's Guide.

For you geniuses here is the...

10 second super-quick-start: Press SELECT to change engines. The right-most rectangles change layers. Use the top LEFT and RIGHT shoulder buttons to change pages. Finally, hold the LEFT directional pad while touching the grid to change sounds.

Installation

Step 1: Unzip cellsDS.zip somewhere on your computer.

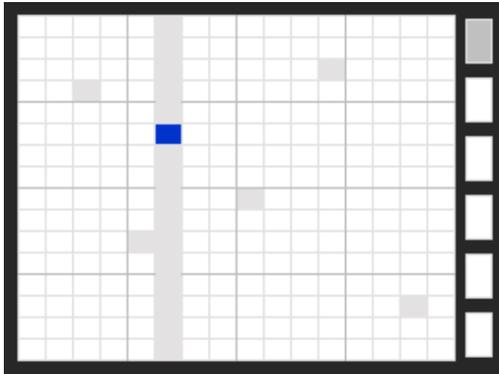
Step 2: Copy the resulting cellsDS folder and all of its contents to the *root* of your card

If you have trouble running cellsDS, make sure that you have installed it to the root of your card. If you are still having trouble, post your issues on the forum at www.glitchDS.com/forum

Using cellsDS

CellsDS is organized into 3 “pages”. Press the top LEFT and RIGHT shoulder buttons on the DS to cycle through the pages. The first and most important page is the sequencer page.

Page #1: The Sequencer Page

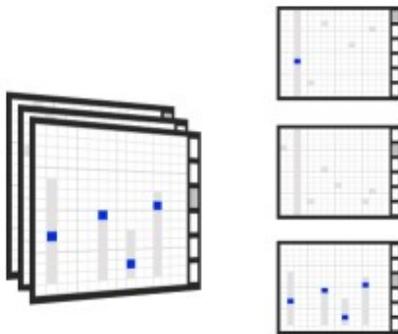


The sequencer page is split between the ‘grid’ (on the left) and the six ‘engine buttons’ on the right. Touching squares on the grid will place and remove notes. If you’re new to cellsDS, your first step should be to place notes on the grid and watch what happens.

CellsDS defaults to the “sequencer.lua” engine which simulates a very simple 16 step sequencer. Later in this documentation I’ll teach you how to choose different engines that offer more unusual sequencing capabilities.

Touching an engine button on the right selects an active layer. All six layers run simultaneously during playback.

Each layer is associated with its own “engine” (which is really just a fancy name for a Lua script). For example, it’s possible to have a song with the following layer/engine combinations:



Layer #1: step_sequencer.lua

Layer #2: step_sequencer.lua

Layer #3: bouncer.lua

Initially, all layers are assigned the engine “step_sequencer.lua”. Choosing a new engine for a layer is easy: Select the *layer* by touching one of the 6 engine buttons, then press [SELECT]. You will be presented with a list of available engines. Select a new *engine* by using the UP and DOWN arrows. Press [A] to load the selected engine into the layer.



(Shown above: Select an engine by pressing the [SELECT] button.)

The technique for selecting sounds might vary from engine to engine. Read the documentation for each engine to fully understand its usage.

Summary of controls for the sequencer page:

- Press the top [LEFT] and [RIGHT] shoulder buttons to switch between pages
- Press [SELECT] to select an engine
- Press [START] to start/stop the sequencer
- All other controls are based on the engine

Page #2: Clock Control

Use this page to control the BPM (beats per minute) of the sequencer playback. The BPM is approximate.

Page #3: Snapshots

To save your current “song” along with all settings, use the Save feature on the save/load page. You may save up to 200 snapshot files. Refer to the instructions on the Snapshots page for details.

Switching Scales

A file called scale.txt in the /cellsDS folder determines the global scale used by cellsDS. The default scale.txt file looks like:

```
22000
24694
26163
29366
32963
34923
Etc..
```

Each entry in scale.txt represents a frequency. In the default scale.txt file, 22000 is the lowest note available in cellsDS and 98777 is the highest note. You can change the frequency values to customize the scale. The following sites might be helpful:

<http://www.phy.mtu.edu/~suits/notefreqs.html>

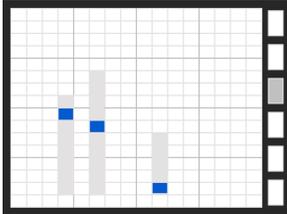
<http://www.phys.unsw.edu.au/jw/notes.html>

About the Engines

The following engines come with cellsDS:

bouncer.lua

bouncer.lua is a simple “bouncing ball” sequencer



- Touch anywhere on the grid to place a new ball or change a current ball’s height. The X axis controls the ball’s pitch. The Y axis controls the ball’s height
- Hold the [LEFT] directional button while touching anywhere on the grid to select a sound. As of this time, not all of the cells contain sounds.

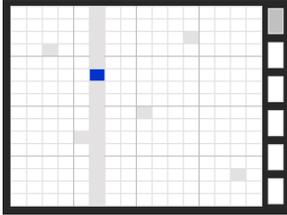


(Select a sound by holding [LEFT] and touching a cell)

- Press [Y] to clear the sequencer
- Touch below a bouncing ball’s column to turn it off
- Hold the [RIGHT] directional button and drag the stylus across the grid to change the panning
- Supports Blocks 1 through 8. (see bs_simple.lua below)

step_sequencer.lua

step_sequencer.lua is a basic 16 step note sequencer



- Touch the grid to place or remove notes
- Hold the [LEFT] directional button while touching anywhere on the grid to select a sound. As of this time, only the top 16 cells (the top row) contains sounds.



(Select a sound by holding [LEFT] and touching a cell)

- Press [Y] to clear the sequencer
- Hold the [RIGHT] directional button and drag the stylus across the grid to change the panning
- Supports Blocks 1 through 8. (see bs_simple.lua below)

strobe_lead.lua

This engine is designed for live play. Hold the stylus on any cell on the first few rows to “play” a rhythmic note. (Try the first and second row for best results.)

- Hold the [LEFT] directional button and touch anywhere on the grid to select a sound. As of this time, only the top 16 cells (the top row) contains sounds.
- Hold the [RIGHT] directional button and drag the stylus across the grid to change the panning

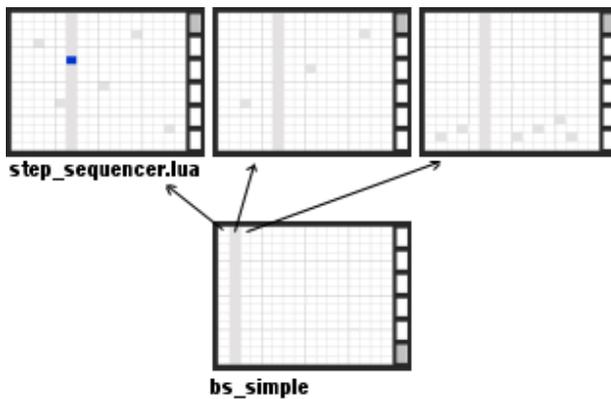
pixel.lua

This engine is an example script for developers. Touch a cell to play a sound.

bs_simple.lua

bs_simple stands for “Block Select - Simple”. Blocks are a big deal. Blocks let you step between a song’s sections: like chorus, verse, etc. When powered up, cellsDS defaults to block #1. To switch to other blocks, you need to load up a script such as bs_simple.lua.

bs_simple.lua is not a sequencer at all, just a block selection tool. Select a block in bs_simple.lua by touching one of the columns with the stylus. Each column corresponds to a block value, namely 1 through 16. The step_sequencer.lua and bouncer.lua engines use this global ‘block’ value to specify different grid configurations.



Technically, cellsDS maintains a global variable named “block” which is shared with all scripts. bs_simple gives you an easy way to select a value (from 1 to 16) for the global “block” value. Since scripts can be user defined, they can choose to use or ignore the global “block” variable.