CDP Composers' Desktop Project

CDP *GrainMill* HELP

GrainMill Reference

- **What is *GrainMill*?**
- **Getting a basic understanding of *GrainMill***
- **How to RUN *GrainMill***
- **Using RANGE to set a group of changing values**
- **SHAPING grains to avoid clicks**
- **TIMESTRETCH to extend the duration**
- **Altering the DENSITY of the sound texture**
- **Adjusting the SIZE of the grains**
- **Location of grains in source file WANDERs about**
- **Moving the grains up and down in PITCH**
- **Controlling LOUDNESS**
- **Randomising output grain timing with SCATTER**
- **Aerial acrobatics: SPATIAL location**

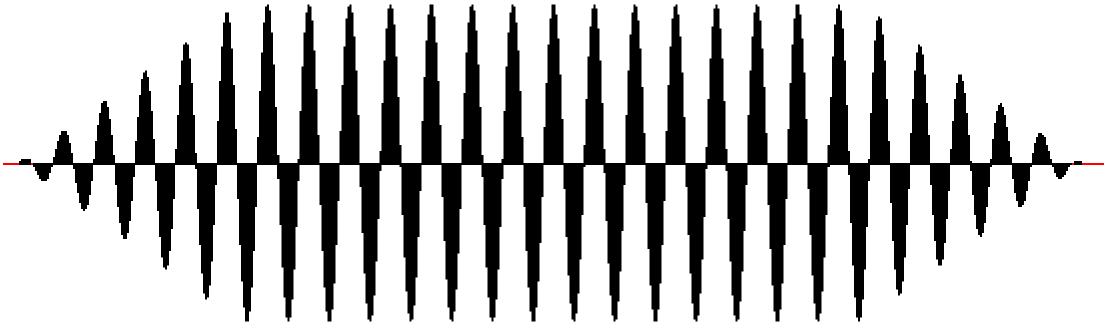
Glossary Items

- **The input soundfile:** the sound with which you begin
- **The output soundfile:** the sound you create
- **What is a 'grain'** of sound?
- **Decibels:** a measure of loudness
- **Milliseconds:** a measure of time
- **Sound-colour:** a description of tone quality
- **Splices:** butt and sloping joins
- **Panning** a sound
- **Mono soundfiles**
- **Default settings**
- **Attack - Sustain - Decay:** envelope shapes

GrainMill Reference

What is *GrainMill*?

GrainMill has been prepared as the first of CDP's powerful programs to have a graphic interface. It is derived from GRANULA, written by Trevor Wishart and available in the CDP System in various versions since 1990. The intention is to make these comprehensive facilities for granular synthesis and sound transformation accessible to a wider range of users, faster and easier to use. We are pleased to acknowledge the support of the Sonic Arts Network, which kick-started this project, the Arts Council of England, which supported the preparatory phase with a generous grant, and the Esmée Fairbairn Charitable Trust, which provided a large grant, enabling us to complete the work.



GrainMill transforms a sound by cutting it into small pieces, called **grains**, then reassembling the grains to reconstruct a new sound. If we simply cut a sound into chunks, then rejoin them end-to-end in the same order, we will end up with the sound you started with! To transform the sound, we need to reassemble the grains in a new way.

GrainMill provides us with several 'new ways' to explore: we can stretch out the original sound in time (**TIMESTRETCH)**, we can alter the density of the grains (**DENSITY**), we can define a 'searchfield' area in the ***infile*** from which to select each grain (**WANDER**), we can change the size (duration) of the grains (**GRAINSIZE**), we can change the pitch or loudness of each grain (**PITCH** and **LOUDNESS**), and we can move them about in space (**SPACE**). Each of these ways to change the sound can be used separately or in combination.

Getting a basic understanding of *GrainMill*

It is useful to appreciate that *GrainMill* is able to reproduce the original sound nearly exactly. It will do this if the DENSITY is at least 2 and no other random factors are called upon.

For example one can use *GrainMill* to perform a simple transposition without changing the duration or other sonic characteristics of the outfileThe_outfile outfile. To do this, DENSITY is set to at least 2, TIMESTRETCH is set to 1, a fixed PITCH Pitch change is selected and no random factors are invoked (e.g., by using RANGE(S)). The original sound will be smoothly reconstructed at a different pitch. A somewhat more sophisticated and effective result would be produced if DENSITY were set to about 8 (or preferably a Range between 4 and 12), and a Range of GRAINSIZES was also used.

You can create a simple granulation effect by setting DENSITY to 1. Now the grains are end-to-end and one can hear the way the SHAPING of the grains causes fluctuations in the loudness of each grain.

True granular effects begin to occur when DENSITY is much higher (e.g, 50 to 200), whereby there will be many grains in the *outfile* where there was just one in the infile. However, to achieve an audible granulation effect, the timing (WANDER) and/or pitch level (PITCH) and/or stereo spread (SPACE) needs to be varied randomly, in the case of the latter two by using them with the RANGE option. Without these random changes, grains are simply copied onto themselves and, however high the DENSITY, the outfile will still sound identical to the *infile*. When each of these many grains is different in e.g. pitch or loudness or source location (WANDER), and the sound is also being stretched out considerably, a complex sonic stream is produced.

GrainMill provides a vast scope of possibilities to explore between smooth reconstruction and massively dense, complex textures in which the granulation effects are dramatically audible.

How to RUN *GrainMill*

RUN topics:

- **A sense of exploration**
- **Settings** for your working environment
- **The Toolbar facilities**
- **Starting up *GrainMill***
- **Processing** with fixed values
- **Activating a range of values**
- **Time-varying** values & BRKEDIT
- **Make** and Play the new sound
- **Continuing** with the existing settings
- **Restarting** with new or saved settings

Explore with an adventurous spirit

Using the program is straightforward, but requires both careful thought and an adventurous spirit, because it offers many and varied possibilities. It is a good idea to explore it freely, trying out various settings and combinations of settings 'just to see what happens' - and observing just what does happen. After a time, you will observe how certain settings produce certan results, and you will then gain more understanding of and control over the results.

System Settings

There are two main options in the SETTINGS dialog box accessible via the Toolbar. These define the overall system environment in which you would prefer to work.

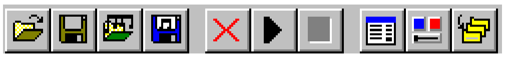
The first of the main options concerns **how you save the soundfiles you make**. This can be done:

- NOT TICKED = 'write directly to named outfile' – You save your soundfiles within the dialog box where you set your parameters. You can accept the default name written in the output filename box ('OUT' is added to the name of your *infile*). Or, you can edit this name to alter it in some way.
- TICKED = 'use the conventional SAVE AS command' (in the FILE drop-down menu) – Note that this is not enabled until after you have made your sound. How can the sound be made if it doesn't have a name yet? A temporary soundfile is made in the directory allocated for this (you can alter this directory in the box provided). The advantage of this second Save option is that you can make sounds, alter the parameter settings and remake them over and over until you're happy with the result **and then use SAVE AS**. The temporary file(s) are automatically deleted when you exit.

The second main option asks you **whether or not you want to 'keep the dialog visible'**.

- NOT TICKED = 'do NOT keep the dialog open when writing and playing sounds' – When you finish adjusting the parameter settings and click on the 'MAKE IT' button, the parameter dialog box disappears. You can then listen to your sound and Save it (if not already saved – if 'write directly to named outfile' is ticked, you have to save the file before leaving the dialogue box). To resume work on that sound, go to FILE and OPEN the same input soundfile as before. Your dialog box, with its previous settings, will reappear.
- TICKED = 'keep the dialog open when writing and playing sounds' – Now when you finish adjusting the parameter settings and click on the 'MAKE IT' button, the parameter dialog box does **not** disappear. You can play back the sound using the 'Play Outfile' button on the Toolbar **and still be able to see what your parameter settings are.** You can even start to make new changes to the parameter settings while you're still listening to the playback. In this way you can very quickly fine-tune your sound, moving easily between playback and parameter adjustments. If you've selected using the conventional SAVE AS command, you don't have to provide a filename until you're completely happy with your sound – and the system takes care of deleting all the temporary soundfiles you've made along the way.

Toolbar



The **Toolbar** at the top of the window which appears when you double-click on *GrainMill* has 10 facilities:

- **Open grnfile** – The 'grnfile' contains all the parameter settings for a particular sound. You yourself (or someone else) will have made this file with 'Save Settings'. See Startup. This is the same as LOAD SETTINGS in the FILE drop-down menu.
- **Save Settings** – You can save the whole configuration of settings used to create a particular sound, including the names of any breakpoint files used, as well as the name of the source soundfile. This saves in a binary file format, and this file can only be opened within *GrainMill*.

  Save Settings saves to the currently named .GRN file.
  Save Settings As prompts you for a new filename.

  NB – When you 'Save Settings' you are asked if you want to associate with the .GRN the soundfile you have been using. Normally, one does **not** do this. When you associate the soundfile, the path to this sound on your hard disk is written into the .GRN file. So it is very specific, and you will not be able to use the same .GRN file with a different sound.

  Therefore, it is more flexible to save the settings *without* associating the soundfile. Then you can use your settings (which includes the names of all the breakpoint files used) with **any soundfile**. Remember to Open a Soundfile **before** Loading a .GRN file which does not have a soundfile associated with it.

- **Save As Text** – To save your settings as a **text file** which you can view and edit, use 'Save As Text' in the FILE drop-down menu.
- **Open sfile** – You select your *infile* and call up the parameter dialog box.
- **Save Sfile** – This save option is only active if you have selected the SAVE AS option in SETTINGS (box unticked). You are prompted to provide a filename for your sound.
- **Stop processing** – You can terminate the production of your *outfile* if, for example, it's going to take a long time and you can already tell that it is not yet what you want.
- **Play outfile** – Audition the newly made soundfile.
- **Stop Playback** – Terminate playback if, for example, it's going to take a long time and you can already tell that it is not yet what you want.

- **Show Dialog** – Restores the parameter dialog box to the screen.
- **Settings** – Define the main options of your overall working environment. See System Settings.
- **Working directory** – Select your working (i.e., 'current') directory. **NB:** If you have saved .grn files and breakpoint files which you intend to use, these need to be present in your working directory. It is recommended to maintain separate directories for different projects, to avoid confusion.

Startup

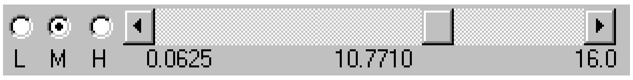
The program is run by double-clicking on the *GrainMill* icon. You then see a title screen with a menu bar on the top. One begins by moving the cursor to FILE and pressing down the left mouse button. A list of options appears. Then move downwards to the option you want and click again. An option is active when darkened. Select OPEN in this way, to open a soundfile with which to work.

> (You may need to change directories to get to the directory and sound with which you want to work. See the select 'Working directory' facility in the Toolbar.)

The existing soundfile which you open in this way and which you will then process with *GrainMill* to make a new sound, is called the 'infile'. The resulting new sound is called the 'outfile'.

> You may also want to LOAD settings saved during a previous working session. You could have done this in two different ways: as a text file or as a (binary) session file.
>
> - If you used SAVE AS TEXT, you need to have a printout or view the file on the screen (in another window) and then use that information to re-enter the settings with the graphic sliders etc.
> - If you used SAVE SETTINGS, you use **Open grnfile** on the Toolbar or LOAD SETTINGS in the FILE drop-down menu to load the binary data file for the project. Select the .grn file you want to use – or select it from the 'most recently used files' portion of the FILE menu.
>
> **NB** – you could have saved the settings with or without its associated (source) soundfile: 'Associate asoundfile.wav with these settings? Yes/No.' If **with**, when you load the .grn file, the soundfile is also loaded. If **without**, you must open a soundfile before you will be allowed to load the .grn file.
>
> You may have had some single or range parameters in place when you decided to use breakpoint files. These single or range settings are also saved and reloaded, but are greyed out when the breakpoint files are selected.

Processing a sound with fixed values



Attention focusses on the slider bars. These are used to select the various possible values. After you open the infile, a large dialogue box will appear, filling the screen. The main menu is still available, awaiting, for example, a different *infile* to be selected without quitting the dialog box.

The processing options are named on the left, starting with TIMESTRETCH at the top of the screen and ending with SPACE at the bottom of the screen. To the right of these process option names are three buttons, marked **L M H** for Low, Medium, and High value spans. You can click on these buttons to set one of three default spans between the lowest and highest values which can be used for that processing option.

The **High** span will display the highest possible value that can be given. The **Medium** span enables you to use the slider bar to select among a span of midrange values. And the **Low** span enables you to move the slider bar through a span of small values.

The value selected in this way is a FIXED value which stays the same as the sound is created.

Activating RANGE to use a range of values

However, it is possible to make sounds in which these values vary randomly within a user-defined RANGE of values.

When you click on the box to the right of the word RANGE, a check mark appears to show that it is now active – and a second slider bar also appears above the original one. It also has the **L M H** span options. Now the lower (original) slider bar is used to set the **lower** of the two values in this range, and the upper (new) slider bar is used to set the **upper** of the two values in this range.

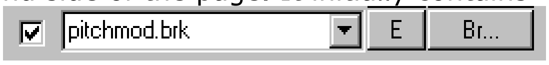
You will notice that the program will not let you set a High value (upper slider) which is lower than the Low value (lower slider), nor a Low value (lower slider) higher than the High value (upper slider). The slider just clicks back to the value in the other slider when you try to move it to these illogical positions. You may, however, see the sliders positioned 'the other way round' if you have selected H (a high range setting) for the upper slider, and M or L (a Medium or Low range setting) for the lower slider.

Time-varying values & BRKEDIT

It is also possible to vary most of the the parameter values over time. This involves creating a 'breakpoint file' which specifies a time and the value for a parameter which is to come into force at that time.

This can be done by writing a breakpoint text file with a text editor. It normally takes the form of separate lines, each with a *time value* pair separated by a space or a tab. It is unusual for any additional text to be allowed, but BRKEDIT does provide an 'extended format' which can handle some additional textual information regarding the type of line span (logarithmic or exponential).

The time-varying section of the *GrainMill* dialog box is on the right hand side of the page. It initially contains default breakpoint files for each parameter. You can enter or browse for the name of a breakpoint text file you have created with a text editor, as above, or you can click on the 'E' (Editor) button to invoke BRKEDIT to create a time-varying breakpoint file graphically.



In *GrainMill* you activate the breakpoint section by ticking the box to the left of the (blank) breakpoint filename box. You can leave the filename box blank or enter a name, using **.brk** as the extension. Now click on the '**E**' (for 'Edit') and the BRKEDIT display screen comes up, with a default breakpoint file in place as a starting point. If you entered a filename, it is recognised and you can simply SAVE the file you make. Otherwise, use SAVE AS, entering the name you want to use.

In *GrainMill* when you return from editing the breakpoint file, having saved it, the filename box will be blank (even if you entered a filename before editing). Click on the down arrow to see the list of breakpoint files associated with that parameter and select the one you want to use.

The **BRKEDIT Reference Manual** is linked to the *GrainMill* documentation, so you can easily access it via the Contents at the head of this file, or via the menu on the left side of the page.

Make – Play – Continue – Restart

MAKE
After the various settings have been made, the sound is created by clicking on the MAKE IT button. You see the main screen fill up with the grains you have made, one mark for each grain, colour-coded for loudness. Sometimes you may unintentionally have settings which will make many thousands of grains and take a long time to process. The **OPTION** to **STOP PROCESSING** is your way out.

PLAY
You can now PLAY the sound you've made. Select PLAY OUTFILE from the OPTIONS menu. If you see that it's going to be very long and you've heard enough, you can **STOP PLAYBACK**.

CONTINUE
If you double click on the infile you have just used (the 'current infile') before exiting from the program, *GrainMill* will restore the dialogue box with all your settings and you can carry on making alterations. This only works for the current infile, and only if you have not yet exited from *GrainMill*.

RESTART
You can select a new infile from the menu while the dialog box is active; this will reset the dialog box to its various Default settings.

SAVE SETTINGS
This will create a binary file with the extension .GRN which contains the data for the sound, parameter settings and breakpoint files used to create your sound. It can be reloaded with **LOAD SETTINGS**.

LOAD SETTINGS
Load the complete project data used in the creation of a sound. These are saved as binary files with the **.grn** extension.

SAVE AS TEXT
This saves the settings in the dialogue box as a text file as a way of documenting your work. It cannot be re-loaded. The binary .GRN file is intended for re-loading.

Using Range to set a group of changing values

The Range option is absolutely essential if good results are to be achieved with *GrainMill*. When you activate this option by clicking on the little box next to the word 'Range', a second slider bar appears for that particular processing function. The lower slider bar is used to set a low limit and the upper slider bar is used to set a high limit for a range of numbers. *GrainMill* then, for each grain, selects a value at random from somewhere between these limits. This grain-by-grain randomisation of values adds the essential qualities of variety and plasticity to the final result - qualities that make all the difference.

Shaping grains to avoid clicks

Creating sounds with sound-grains raises the question as to how the grains are joined together. You can run *GrainMill* in such a way that it creates audible gaps between grains a staccatto or hiccuping effect. However, if the new sound you make is to have no silent gaps between the grains, it seems obvious that the grains must either overlap one another, or 'just touch' one another. In practice, however, the grains must always overlap just a little bit, to produce a continuous output sound. Why is this?

It is not possible simply to chop up sound as if with a knife. If you do this, you will produce a click where the cut sound ends, because the volume is likely to fall off a cliff, as it were: from high to nothing within an instant. Among other things, the physical mechanism reproducing the sound for our ears cannot cope with such an abrupt change. To avoid this, we use a tapered cut, called a 'splice slope'. A splice crescendos from silence at the start of the sound (its **'attack'**), or decrescendos into silence at the end of the sound (it's 'decay') though **butt joins** are also possible (no sloping envelope at all). In this case, the 'sound' is a single grain. However, we usually don't want to hear these little crescendos and decresendos in our new sound because of the tremolo effect produced by the (very) rapid fall-rise-fall-rise etc. in volume. It is for this reason that the program overlaps the grains a little, so that the crescendo on the following grain comes in as the previous grain decresendos, thus maintaining a certain consistency of volume.

Stretching time with *GrainMill*

TIMESTRETCH alters the length (duration) of the outfile.

When the TIMESTRETCH slider bar is set to a value greater than 1, grains which overlap are cut from the *infile* and then reassembled in the *outfile* in such a way that they do not overlap as much, thus stretching out the sound.

When the TIMESTRETCH value is 1, there is no change.

When the TIMESTRETCH value is less than 1, grains which do not overlap are cut from infile (there may even be gaps between these grains: bits of soundfile missed out) and then reassembled in such a way that they do overlap, thus shrinking the sound.

The upper limit of TIMESTRETCH can be very high and should be used with caution.

The 'outfile duration' box at the bottom of the dialog box shows the effect of changing both TIMESTRETCH and GRAINSIZE. The duration is a maximum estimate, and can be edited if necessary.

In fact, the outfile duration box is active not just reporting what the program is doing.

A special case

For example, it is necessary to provide (i.e., type into the outfile duration box) a duration in the special case when TIMESTRETCH is set to 0. With this setting, the initial grainsized segment of infile is used over and over, possibly forever - hence the need to specify a length for the outfile. In this special case, the infile is being used simply as the source of a waveform of some kind: the source from which the grain is to be extracted. In the simplest case, this infile need be scarcely any longer than the required grainsize.

The simplest grain waveform is a sinewave of arbitrary pitch. However, any input sound can be used, including Frequency Modulated (FM) waves, and chirp waves (a sine wave of rapidly increasing pitch), as well as acoustically sampled waveforms.

Altering the density of the sound texture

When DENSITY is 1, the grains are placed end to end (a **butt edit**) and the original sound pulsates slightly at a speed related to the grainsize.

As DENSITY becomes increasingly greater than 1 (more dense), the grains overlap more and more, producing a continuous to smooth/dense sound. For example, when DENSITY is set to 5, there will be 5 grains in the *outfile* where there was 1 grain in the *infile*, i.e., 5 grains overlapped where previously there was only 1 grain. When one or more of the RANGE options are activated, each of these 5 grains will be separately processed. This is how the *outfile* becomes more and more complex. Densities of 50, 100, or even 200 simply make the outfile texture more and more complex depending on the other settings. The upper limit in this program is 256.

As DENSITY becomes increasingly less than 1 (less dense), we begin to hear silences between the grains. If the grains are very short, and the density is only just below 1, this will appear as a kind of grittiness in the new sound. But if the density is nearer to 0, the new sound will appear as a stream of tiny, separate **attacks** or grains.

Adjusting the size of the grains

The size (i.e., duration) of each grain has a major effect on the result. In the first place, the fact that each **grain** of sound has splice slopes sets some limits which cannot be over-stepped. The fact that we need two of these splices for every grain (one ascending from silence and one descending to silence) limits the size of the grains we can use: the smallest grain cannot be shorter than the two splices don't worry, the program makes this impossible. The smallest grain allowed by the program is currently 12.5 **milliseconds** (0.0125 sec), and this provides the space needed for the splice slopes at the beginning and end of each grain.

Furthermore, if the grains are very, very, very short, they will contain so little information from the original sound that its qualities will be destroyed. So there is a lower limit on the size of the grains. If your grains are about 50 milliseconds in length (0.05 of a second), they will happily capture the qualities of your sound.

Once the grains become much longer than this, we create a different effect. Almost every natural sound changes through time: it changes pitch, loudness, and its sound-colourSound_colour. If you are using grains to timestretch a sound to double its length, the grains you cut from the original sound will use overlapped portions of the original sound. In fact, each grain will contain a copy of half the previous grain.

If the original sound is itself changing, and your grain is sufficiently long, the sound inside the grain will itself be changing. When you now place the next grain after the first, you will hear that change in the sound being repeated .. the output sound will seem to backtrack to a place earlier in the change-process. There will be a kind of very fast echo inside the sound.

These 'internal echoes' may be a very interesting effect if this is what you want to hear. However, if you just want to make your original sound longer, you need to avoid this internal echoing. It turns out that a grainsize of around 50 milliseconds (0.05 sec) captures enough of your sound to retain its special sound quality, but not so much of it that you are able to hear any change in the sound within the grain. Using grains of about 50 ms allows the sound to be stretched without any sense of internal echoes.

Grainsizes longer than 50 ms allow more of the original sound to come through. A sandy beach reveals very little of the shape and quality of the original stones from which the sand was ground, but shingle is much closer to the original shape and texture of the rock from which it was formed.

Note that there is a relationship between the type of input sound and the effect a given grainsize will have. The sharper (faster) the attack transient in the infile, the more it will be present in the grain, even with a fairly short grainsize, and certainly with a grainsize as long as or longer than this attack transient.

Using a range of grainsizes

The RANGE option enables you to set a range of grainsizes. When activated by clicking on the box next to the word RANGE, a second slider appears. The lower slider is used to set the low limit of the range, and the upper slider is used to set the high limit of the range. Now the program will pick out grainsizes at random between these two ranges, a different grainsize for each grain. Introducing this random suppleness into the texture improves the result considerably: it is more lively and feels more natural.

A very good range to choose is between 50 ms and 80 ms. Ranges greater than 100 ms (0.01 sec) will begin to produce collage-type textures, i.e., sounds in which portions of the infile are distinctly recognisable.

On internal echoes

Echoes inside the new sound can be useful. Room reverberation is produced when the sound waves from some event in a room bounce back off walls, floors, ceilings and other surfaces, producing a multitude of slightly different copies of the original sound - each arriving at our ears at a slightly different time. There are so many of these different echoes that we hear them not as true echoes, but as reverberation, which enlivens and enriches the original sound.

If you:

- choose grains not quite in exact sequence (by using WANDER), and
- place them in slightly randomised positions in the output sound (by using SPACE), and even
- use a slight microtonal variation of pitch (by using PITCH), and
- make small variations in loudness (by using LOUDNESSS), and then
- create a higher density (by using DENSITY)

... then you will produce an effect like the reverberation of a room.

Once your grains start to become any larger than 50 ms (0.05 sec), you will generate obvious echoes inside your reconstructed sound. If your grains are 'very' long (e.g., 1 second) and if you use the WANDER process over a very large span, you will produce a kind of random collage of pieces of your original sound.

Wandering grain locations

WANDER defines a length of soundfile within which the next grain is to be selected at random. The longer the length, the more likely it is that the new grain will come from a location ever more further away from the previous grain. This length is sometimes called a 'searchfield', i.e., the span of sound to search when selecting a grain. A small searchfield to 'wander' about in will result in a somewhat blurred image, whereas a large searchfield will result in scrambling the original sound.

With WANDER set to 0, grains are selected from *infile* in the strict order in which they were cut in the first place.With WANDER set to (say) 0.1 sec (100 ms), a grain is chosen from *infile* within a range of 0.1 seconds before the 'true' position. In other words, there is a certain vagueness about the order of the grains chosen for use in your new sound. The larger the WANDER, the more vagueness there is!

And if you set WANDER to cover the entire length of the original sound, then the vagueness will be complete: your original sound will become completely scrambled. It is useful to know that the searchfield gradually expands when it is set to the length of the entire soundfile. *GrainMill* reads through the infile, and as it does so its 'here' position is moving through the file. The searchfield is from the beginning of the file to 'here' and thus gradually gets larger and larger. This means that the scrambling effect will be somewhat minimal at the start and then increase steadily. This transitional process from smooth to chaotic can be useful musically.

Moving the pitch of grains up or down

Each individual grain can be altered in pitch. When you set PITCH to a (fixed) position along the pitch span slider bar, all the grains are transposed by this amount.

The span of pitch change values is roughly based on 12, so that each integer change represents a transposition of more or less 1 semitone.

Values less than zero (i.e., negative values, such as -0.5) transpose the grains downwards in pitch. Remember to precede the number with a minus sign!

Note that altering the pitch also alters the length of the grains (higher pitch shortens grains, lower pitch lengthens grains. However, if you set TIMESTRETCH to 1, the duration of the *outfile* will remain the same as the *infile*, because of the way the grains are reassembled.

When you activate the RANGE button, a second slider appears and you can set lower and upper limits to a range of transposition values. The program will then, for each grain, select a transposition value at random from somewhere within this range. The sonic results of these random jumps of pitch can vary remarkably as the size of the range is increased.

Normally, an especially fast method is used to transpose a grain. If a large value is used, especially for downwards transposition, this can lead to unwanted grittiness or distortion. This can be prevented by clicking on the 'HiFi Transpose' button, which tells *GrainMill* to use a slower but much more accurate method..

Controlling the loudness of the grains

Each individual grain can be altered in loudness (also known as 'amplitude'). When you set LOUDNESS to a (fixed) position along the loudness span slider, all the grains become correspondingly louder or softer. The LOUDNESS span represents **decibels** (dB) and extends from silence (-60dB) to full (original) volume (0 dB). The action of the tapering grains and splice slopes of each grain will have a volume-reducing effect, but the program does its best to minimise this and prevent the sound from becoming too soft.

GrainMill uses a default splice slope of 5 milliseconds. This gives a fairly sharp attack (and decay) for each grain. If you want to soften this attack, you can select one of the other splice slope options provided. Long grains with long (gradual) slopes makes sense, but putting a long slope on a very short grain will either not be allowed or may reduce the output volume by an unacceptable amount.

When you activate the RANGE button, a second slider appears and you can set lower and upper limits to a range of loudness values. The program will then, for each grain, select a decibel level at random from somewhere within this range. Having a range of random loudness values in the texture adds a great deal to the character of a granular sound.

Timing scatter of output grains

TIMING SCATTER is adjusted by moving the slider control. TIMING SCATTER creates small random fluctuations in the timing of the grains in the soundfile created by *GrainMill*. This may produce a more supple, natural feel in the flow of the grains. If it is set to zero, pitch artefacts may result from the regular timing of the grains. Trying it both with and without SCATTER may be necessary to determine what best suits the sound and the context. The control comes up with a default setting which may be retained until a reason do otherwise becomes apparent.

Aerial acrobatics: spatial location

Here we are processing the movement of sound through space: to be more precise, the Left < > Right movement of a stereo image across a horizontal plane, known as **'panning'**. The SPACE process re-positions each grain in the horizontal stereo field.

If the input is **mono** (one channel), the program will automatically produce (the necessary) stereo output when SPACE is used. The setting on the SPACE slider bar selects a fixed position in the Left-Right stereo field.

When RANGE is activated, a second slider bar appears and a specific range can be selected, usually using the lower bar for the Left limit, and the upper bar for the Right limit.

Full Left on the lower slider bar + full Right on the upper slider bar sets the full stereo field as the range within which each grain is randomly positioned. This can be tightened by moving the sliders more to the middle.

Sounds panned between full Left on the lower slider bar and Centre on the upper slider bar means that only the left side of the field will be used.

Sounds panned between Centre on the lower slider bar and full Right on the upper slider bar means that only the Right side of the field will be used.

If both sliders are set to Centre (the **defaults**), the program will produce a mono *outfile* (making it pointless to use the RANGE option). Note that any non-zero SPACE value will cause the program to produce a stereo *outfile*. However, the sound can be located on only one channel of this stereo soundfile if either:

> a single (fixed) value to the Left or Right of Centre is used,
> OR
> both sliders are close to the Left or Right.

It is the distance between the sliders that sets the stereo spread of the sound.

Glossary Items

The infile

The infile is the currently selected soundfile to be processed. It must be a mono soundfile(one channel). As with any sound processing, the qualities of the infile greatly influence the range of possible results.

The outfile

The outfile is produced by *GrainMill*. It will be mono or stereo depending on the options selected. The program automatically provides a name for the outfile by adding 'OUT' to the name of the infile. To edit this name, move the cursor to the outfile box and click to place an active vertical line cursor in the box. Move left and right with the arrow keys, deleting to the left with the Backspace key, and to the right with the Delete key. For example, if your infile was called 'gromont.wav', the default outfile name would be 'gromontOUT.wav'. You could name a second version, 'gromontOUT1.wav', etc. (Filenames over eight letters in length are possible with the Windows 95 operating system for which this program has been written.)

What is a 'grain' of sound?

A 'grain' is a tiny bit of something - in this case sound rather than e.g., sand or corn. A sound made up of lots of small 'grains' usually has a somewhat rough texture. It can sound just a bit rough-textured or bumpy, or may sound fiercely repetitive. It all depends. Similar to the paintings of Seurat who used tiny dots of paint rather than the usual brush stroke, the sound is made up of lots and lots of tiny pieces. A sound made up of grains is described as 'granular', and creating sounds like this is called 'granular synthesis'.'

Decibels: a measure of loudness

The scale of loudness calculated in decibels (abbreviated as dB) is a logarithmic scale. The logarithmic scale is used for decibels because this is how the ear works. What this means is simply that equal divisions of the scale will sound to us as equally louder or softer, whereas mathematically the values are actually doubling. Logarithms are a way of linking linear and geometric values (such as 1, 2, 3,4 and 2 4 8, 16): a single (i.e., equally spaced) step in the linear scale involves a doubling of the previous value in the geometric scale. If the loudness scale were to be represented on the screen as a linear scale (and the same is true of pitch), most of the changes that we can perceive with our ears would be bunched together at one end of the scale. Textbooks on acoustics provide a full mathematical explanation of these matters, and often include a chart illustrating where familiar sounds normally come on the scale of decibels.

Milliseconds: a measure of time

A millisecond (abbreviated as ms) is one-thousandth part of a second. The thousandth decimal place is the third to the right of the decimal point (tens-hundreds-thousands). So 1 millisecond can also be written as 0.001 sec, 50 ms can be written as 0.050 sec, and 100 ms is the same as 0.1 sec. To convert from milliseconds to seconds, move the decimal point three places to the left (adding zeroes if necessary), and to convert from seconds to milliseconds, move the decimal point three places to the right.

Sound-colour: a description of tone quality

The 'colour' of a sound is more technically known as its 'timbre'. This means how many and which frequencies are present as the sound flows through time. This frequency content is constantly changing, in some sounds more than in others. For example, a flute has a clear luminous tone, while an oboe has a considerably more complex sound, and a train engine has a very rich sound indeed! We have adopted the word 'colour' from painters to describe these differences; they all come down to the ever-changing frequency content of a sound.

Splices: Butt and sloping joins

In the 'classical tape studio', analog tape was spliced by cutting off selected pieces with a razor blade and then joining up the pieces with a special white splicing tape. These cuts were usually made with the help of a splicing block, and there was a groove at 45% for a 'smooth' join (the two pieces of tape would overlap) and a vertical groove for what was called a butt edit (no overlap at all). Similarly, in *GrainMill*, the grains are 'abutted' when there is no overlap. However, note that there is in this case still the tapering of the sound from and to zero volume, with which each grain is shaped in order to avoid clicks.

Panning a sound

The terminology 'to pan a sound', 'panning a sound', or just 'panning', refers to the psycho-acoustic effect whereby sound appears to move through space. To illustrate: a (stereo) sound can 'move' from left to right. In this case, the stereo image is actually coming out of both speakers, but the right hand speaker starts with zero volume and the left hand speaker with full volume. Then the right hand speaker gets progressively louder as the left hand speaker gets progressively quieter. To the ear of the listener (positioned centrally in front of both speakers), the sound appears to move from left to right.

Mono Soundfile

A mono soundfile has only 1 channel. If there are two loudspeakers, the(same) sound signal is output by both of them.

Defaults

Programs frequently use 'defaults'. A Default is a pre-set value, particularly for frequently used settings. It can usually be changed by the user, but it can be convenient to have it there automatically to start with. It saves having to set every single value that is needed.

Attack - Sustain - Decay

These terms refer to the loudness shape (amplitude 'envelope') of the sound. They describe some of the most noticeable and important features of a sound. The attack transient is how the sound begins, in particular how quickly the loudness increases; the Sustain portion refers to the loudness characteristics of the main part of the sound the Decay portion is how it ends, for example whether it stops abruptly or gradually trails off into silence.

Last Updated 3 August 2015
Documentation: Archer Endrich, revised R. Fraser