

“One: demonstrations always crash. And two: the probability of them crashing goes up exponentially with the number of people watching.”

– Steve Jobs

(Founder of Apple Computer, Inc.)

Testing and responding to feedback is one of the most important stages in the development of a software product. Although VSTrack is not a commercial enterprise, one of its goals is to appeal to as many musicians and computer users as possible. Software development in tracking is often the story of a musically-minded programmer who looks to create a program for their own use – without regard to the needs or wants of other people. As such, their software is only likely to appeal to people of a similar mind – namely other musically-minded programmers.

VSTrack has inherited the interface of one of these previous trackers on the basis of its *efficiency* – and, whereas VSTrack v0.84 (as we have implemented in our project) does not implement all the user interface improvements we identified in Chapter 3, it is now time to reflect on the two important traits of user-friendliness: ease-of-use and ease-of-learning.

In this chapter, we take three approaches to evaluating VSTrack. Firstly, the author offers a brief review of the alpha testing and debugging stage, together with a subjective response on the resultant program. Secondly, we record the findings of an Internet-based release program, designed to gauge the objective response of the online technically-proficient community. Finally, we report on a seminar that took place in Trinity College, Dublin, conducted to measure the response of the more musically-inclined audience of VSTrack.

5.1 subjective response

Before offering a response on VSTrack v0.84, there are a couple of points to note. Firstly, though v0.76 subsequently prompted the beginning of the alpha testing stage, there was a considerable amount of testing done during the actual coding – at all stages of development. The modular approach to programming that C++ offers often permits the testing of components as they are individually completed. Thus, all programming ‘Gestalts’ at some stage will have been independently tested: from the performance of the core GUI sub-system, to the robustness and effectiveness of each individual pattern effect command (see Appendix A).

Secondly, it should be noted that v0.84 is an *alpha version* of v1.0. Though it has been our intention to produce a mature and stable product for this critique, we make no claims to the market-readiness of the current version of VSTrack; in no way should it be considered a benchmark for future tracker / sequencer hybrids. Indeed, several desirable features (previously discussed) are yet to be implemented, and a continued period of testing (notably *beta* and *final testing*) remains to be conducted.


alpha testing

As mentioned, testing during development largely revolved around individual features, rather than the cohesiveness of the product, as a whole. However, the beginning of the alpha testing stage marked a point where the features began to mature, and it became necessary to test the program in its full capacity: as a program for writing music. Fortunately, the demo tune required for the seminar (section 5.2.2) provided an excellent trial for the program.




Happily, this author can report that teething problems in this capacity were minimal. Most issues involved specific instances of pattern effect commands interfering with each other – notably, conflicts between panning slides at the note and channel levels. Other amendments included: the refinement of the module loading algorithm (to tolerate illegal and unknown XML tags, and allow objects to be specified in the file in an arbitrary order); the refinement of the sample loading algorithm (to record and adapt the original *Microsoft Wave* file header, so that it could be attached to the module's own samples); the patching of miscellaneous memory leaks (notably, of `COffscreenContext` objects, which had a tendency to make the host unstable). However, after only a few days of intensive testing, the program was stable enough to compose the demo with little or no problems.

subjective response

In this author's opinion, VSTrack v0.84 successfully satisfies the objectives laid down in section 3.2.1. In comparison to IT2, VSTrack v0.84 has managed to retain much of the interface traits of its ancestor – making it efficient, flexible, powerful and easy-to-use. Also, though channel capacity (polyphony) has dropped, due to increased CPU usage, the audio quality has significantly improved⁷¹. Graphics performance, particularly in the pattern editor, is also swift – the editing process gives an impression of quick reaction and response. After the initial alpha tests, this author was also surprised by the general stability and robustness of the application.

However, though the program works well within itself – and synchronises effectively and accurately to the host (*Steinberg Cubase SX*) – there are a couple of issues relating to input mediation between the plug-in and Cubase. Firstly, an interesting issue comes in the form of the  key, which is used to change tracks in the pattern editor, and to change control focus elsewhere. When the

⁷¹ Sample playback in VSTrack is of very high quality. However, there are some minor glitches associated with abrupt changes in certain note parameters – known as *parameter noise*. These artefacts should be relatively easily addressed, in VSTrack v1.0, with the application of a low pass filter (LPF) to parameter changes.

user first uses the key, Cubase – instead of supplying the input to the plug-in's handler – handles it itself. As such, the control focus instantly changes from VSTrack to the text box atop the VST plug-in window, denoting the name of the current plug-in preset (unused in our application). The user is forced to press the  key to return to the plug-in. Unfortunately, there doesn't appear to be a workaround that can be effected by the plug-in itself – this remains an issue for Steinberg. Fortunately, however, this only occurs once in every session; after the user has pressed , the  key functions as intended.

Also on the subject of focus, when users are moving between the sequencer and the tracker module, they are required to first move *window* focus from one to the other (to determine which should handle keyboard input). Unfortunately, the best way to do this at the moment is to use the *mouse* to click on the appropriate window – something we wish to avoid. Preferably, it would be better to have a shortcut key available in the host that will instantly take the user to VSTrack, and one in VSTrack that will return them directly to the host. Again, however, this is something beyond the purview of plug-ins.

In closing, we also identify room for improvement in the area of memory usage and performance. In the former case, memory efficiency has been sacrificed for graphical performance. However, this has yielded a plug-in that demands 20-30 megabytes of memory for offscreen graphics buffers. Though memory is usually abundant in modern computers, it would be desirable to find a more economical compromise between the two – indeed, it is conceivable that a negligible drop in graphics performance, while most likely going unnoticed by the user, might yield more pronounced savings in memory.

As regards audio performance, though we do not have the benefit of assembly language or hardware and though we also concede a performance hit for the sake of audio fidelity, it would be desirable to decrease overall CPU usage – potentially allowing us to increase the number of channels, to compare more favourably with other trackers (e.g. 32 or 64). To this end, two possible solutions can be identified: the conversion of all floating-point numbers to integer representations, or the possible application of *Steinberg's ASIO* interface [61] for better communication with hardware.

5.2 objective response

The developer's perspective can only offer us so much in the way of feedback. The ultimate test of a program – notably one that relies on its user interface [49] – is the *user test*. In bringing two technologies together, we also hope to combine their respective communities. In the following sections, we report on two different approaches, engineered to elicit each point of view, respectively.

5.2.1 internet testing and feedback

To assess the program's reception by its technologically-proficient audience, the Internet is used for the purposes of releasing the binary (executable) VSTrack program. To this end, the plug-in was compiled and combined with brief documentation⁷², then made available for download on the author's website (<http://www.nashnet.co.uk/>).

soliciting feedback

An issue that had to be overcome, however, was determining the method (and incentive) for feedback. The Internet is seen by most as a place to download or receive information, and only seen by a minority (albeit sizeable) who actually upload or contribute themselves. As such, a way had to be found to encourage people to actually provide feedback, once they had downloaded and run the program.

Though web server usage statistics can be used to find out how many people have downloaded a particular file on a website, the amount will represent not only those who are actively interested in VSTrack, but also today's large number of surfers who will download anything and everything – possibly never to actually give it any due attention.

alpha release
procedure

The release was therefore conducted as follows:

1. VSTrack v0.84 was presented for download on a page of the nashNET website. However, to access this page, users must first enter their name, email address, list any proficiencies that might have (e.g. music, programming, sequencers, trackers, etc.) and agree to the terms and conditions of download. These two initial pages are presented in Appendix D.1(i) and (ii), respectively. Rather than advertising it as an "Alpha Download", emphasis is given to the venture as an "Alpha Testing Programme", to which participants must first apply.
2. At several stages in the documentation, the participants are reminded that feedback is expected. The notion of *reciprocity* [11] is exploited to implicitly suggest that since the developer has provided the user with a program, the user is 'obliged' to provide feedback in return.
3. The existence of the website (and release) is advertised in suitable forums only: the VST developers mailing list and appropriately themed newsgroups (e.g. MIDI-, tracker-, sequencer- and Steinberg-related groups). In addition to targeting experienced VST and sequencer users, this will also target people familiar with the technology and process of programming VST plug-ins.
4. To make feedback as easy as possible, users are directed to an online form, on the nashNET website, that solicits responses on particular aspects of both the concept and implementation of VSTrack, as well as sampling more general opinions about computers, sequencers and trackers. This form is illustrated in Appendix D.3. Users are also given the alternative of more direct and informal feedback through email.

⁷² Accompanying documentation included a `readme.txt` file: which contained a description of the program, brief instructions to its loading, copyright and legal information (including a disclaimer) and a list of "known issues" – problems or "undocumented features" that were already known, but have not been (or cannot be) solved, at the time of release. This file is reproduced in Appendix D.2.

Internet response

The release was highly successful. Though the targeted newsgroups seemed to yield little return, the VST mailing list provided a substantial supply of interested parties. In the first three days, over 2000 people had already visited the VSTrack website – and over 300 people had downloaded VSTrack 0.84. Though attendance and interest dropped as the announcement post was buried by newer forum messages, the first 10 days had brought 3,491 visitors and produced 619 applications to the alpha testing programme. It should be noted that the discrepancy between visitors and downloads cannot be interpreted as a reaction to the VSTrack concept – it is quite likely that many visitors, though interested and willing to download the program, were unwilling to supply an email address in order to do so. Indeed, it is estimated that about 15% of those who did apply nonetheless supplied a phoney address.

Feedback, as expected, also represented another decrease in numbers. However, at the time of writing, 45 people have already submitted their comments and suggestions (split roughly halfway between the online form and direct email contact). Their general feedback is summarised as follows:

Internet feedback

Re: VSTrack. The concept was very well received; some commended the originality of the idea, while others said they had been eagerly awaiting a VST tracker plug-in for some time. One former IT2 user said, “I felt so familiar with this program, like it has always lived in my computer, and it kind of pulled me forward to sit in front of the screen and focus on overcoming the limitations of the host.”

Another user exclaimed, “[The] idea of the tracking approach in parallel with other technologies like Digital HD recording and MIDI is excellent... finally we can have the best of all composing techniques (Score, Audio, Sampling, MIDI and now finally the tracking approach) inside a music application like Nuendo or Cubase, which will represent *the studio*.”

Most (76%) could see themselves using VSTrack in the future. Though all feedback seems to commend the concept, some people did run into difficulties. Notably, since VSTrack had not been tested with VST hosts other than Cubase (as had been forewarned in the release), there were a number of participants complaining of incompatibilities with other VST hosts. For example, Logic appears to have loaded VSTrack perfectly, but will critically not allow any keyboard input – as was the story in some other hosts. Impressively, VSTrack did seem to work well in *Vsthost*, a standalone freeware VST host.

As was the view of this author (see section 5.1), a number of people would like to see VSTrack v1.0 capable of playing back more concurrent tracks, but themselves accepted that top-notch performance is not a priority in *alpha version* software. However, a couple of people – on slightly dated platforms – did exhibit more serious problems owing to the large memory requirements of VSTrack; in one unfortunate case, the computer was unable to load the fonts, and thus could not present the pattern data on the screen. Most users, however, had no issue with stability or reliability.

One person suggested a departure from the “old-school” (ST3/IT3) aesthetic, and many would like to see mouse support in subsequent versions, but another said, “Mouse support would be good for a few things, like changing the panes instead of using F2 to F3 and so on, but it’s great that you force people to use the keyboard actually.” Some also commented that the font size was too small for people working at high-resolutions – as is recommendable for modern sequencers (see section 2.3.3).

Re: Computers. Nearly all (98%) of participants had significant experience with computers – many of whom claimed to be expert in their use. Preferences for operating system were unsurprisingly Windows XP⁷³. Preferences for input devices placed the keyboard as the first choice of this audience (mostly ‘power users’), followed by the mouse and the MIDI keyboard, but with occasional votes in favour of the graphics tablet (pen or stylus).

Re: Sequencers. Most (87%) participants had previous experience with sequencers – many of whom claimed to be proficient in their use. Preferences for specific sequencers was biased towards various versions of Cubase and Logic – but familiarity and limited exposure to alternatives seem to have significantly contributed here. In some cases, Cubase users cited stability as a factor that tipped the balance, but many conceded that the lack of diversity in modern sequencers – coupled with the time it takes to learn such programs – did not encourage migration between platforms.

Re: Trackers. Roughly half (48%) of participants had previous experience with trackers. Of these, most were in full appreciation of the advantages offered by both the keyboard-bias and interface. Preferences for specific trackers varied, but Renoise currently appears the most popular – with people attracted by its impressive feature set, audio capabilities and performance. Several also cited fringe-tracker, FL Studio (Fruity Loops) – an intuitive mouse-based pattern-based step-sequencer.

closing remarks

In section 6.1, we shall discuss the lessons learnt in this release, while highlighting future work that might be prompted by it. However, before proceeding to an appraisal of the seminar, it is interesting to note that a significant proportion of the Internet audience came not from the VST mailing list or targeted newsgroups *directly*, but from referrals made by other sites, who took it upon their own initiative to advertise VSTrack. Notably, almost half (40%) of the participants were referred from www.kvr-vst.com – a very popular site devoted to reporting the news on new and interesting VST plug-ins, which covered VSTrack’s development with keen interest. Additionally, almost a dozen independent discussion threads have appeared across the Internet, in relation to VSTrack – including sites in Germany, the US, Finland and Japan, as well as www.united-trackers.org, which we mentioned earlier (section 2.3), and the *Renoise* website forum (www.renoise.com). Though this enthusiasm is highly gratifying, it should be noted that it may have skewed the target sample slightly – from VST developers to VST users. This, however, in no way invalidates the feedback, summarised above. Furthermore, it is expected that such a trend will continue, since two magazines, *Computer Music* and *Future Music*, have recently indicated an interest in covering VSTrack’s final release.

⁷³ Though ‘familiarity’ was cited by most participants, it should be noted that the release of Windows™ software will always attract Windows™ users, before those of any other platform. Though this might seem like a moot point, it is interesting (even if not conclusive) to gauge the preferences of visitors from the VST developer list, who will often run both Windows XP and either Mac OS9 or OSX. It is perhaps significant that in many such responses, Windows XP was the preference.

fig.5.1 – the author,
introducing VSTrack



Internet surveys make several assumptions about their sample audience: that participants have Internet access; that they have their own computer to run VSTrack on; and that they have the technical knowledge to install and run the software without supervision. Indeed, a considerable amount of software distribution in the modern age takes place over the Internet; a product must be designed with such channels in mind. However, our desire is to assess VSTrack – notably in relation to its operating interface – rather than assess VSTrack's installation procedure.

target audience

To this end, on Thursday, the 4th of June, 2004, VSTrack v0.84 was demonstrated to several members of the Music and Media Technology (MMT) course, in Trinity College, Dublin. Sadly, the timing necessarily coincided with various deadlines for the students of the course. Nonetheless, there were eight in attendance, at the seminar – comprising a healthy cross-section of students (from all levels – diploma, masters and PhD.) and staff.

In addition to making the software more accessible (the workstations having been previously prepared), the personal milieu also facilitated more direct and informal feedback; any queries, comments or problems could be instantly addressed or freely exchanged. Though it should be noted that the presence of the software's developer might have indirectly tamed some of the responses.

One of the strengths of the MMT course is that it draws from a wide range of disciplines – from musicians to technologists and everything in between, including architects, artists, physicists and many other fields. The only unifying quality of all its applicants is an interest (though not necessarily proficiency) in music (or media) and its technology. In this way, the seminar was able to sample the response of a more musically-oriented audience – without excluding those of a technical inclination. It should, however, be noted that the technical proficiency of the students in attendance – owing to the course itself – will have recently

improved. Though this would appear to again skew the findings of the seminar towards that of a technically-persuaded sample, any recent newcomers to computers (and music) will nonetheless be better able to empathise with the novice user – in some cases their former selves.

seminar
programme

The seminar proceeded as follows:

1. The first 5 minutes were given over to an audio / visual presentation, which welcomed the participants and proceeded to play a prepared demonstration song (composed using VSTrack, with Cubase SX), accompanying several screenshots of the application as the music played. The presentation is supplied on the CD-Rom, accompanying this report⁷⁴. The demo song is also provided on the CD-Rom, in audio format – playable on any audio CD player.
2. An intentionally brief introduction (fig.5.1) was then given by the author, guiding participants to the VSTrack application, on the computer workstations before them (pictured in fig.5.2). Participants, having been provided with a brief tutorial (as might be provided by a minimal soft- or hard-copy manual – given in Appendix D.4), were then encouraged to use the programme for a short period, without active guidance from the author – so as to as best emulate the real-world situation.
3. After the initial hands-on session, a brief demonstration was given to the audience, detailing the finer points of VSTrack. Using a computer connected to the room's projector, the demo song's module and sequencer files were opened and explained (see fig.5.3). This included a step-by-step description of the methods used to create the synthesizer texture (heard at the start) and dense rhythmic patterns (heard throughout), followed by an explanation of some of the more powerful tracker effect commands (notably, the sample offset command, Oxx – see Appendix A) and features of trackers that might make them more adept to certain musical tasks, in relation to sequencers and conventional methods.
4. Participants were then encouraged to return to the software before them, while the author responded to questions and comments on the concept, workings and potential of VSTrack – sometimes in a one-on-one capacity, but more often as part of a group discussion.
5. Before leaving, attendees were asked to complete a survey (also given in Appendix D.4), which paralleled that of the Internet trial, but also allowed for a freer (and potentially anonymous) method of contributing feedback during the seminar.

⁷⁴ The presentation was engineered in *Macromedia Flash*, but the supplied version has been converted to an self-executable format. The reader should be able to simply run the file, without concerning themselves with the installation of any dependencies, such as the *Macromedia Flash Player*, etc.

fig.5.2 – example of the workstation for seminar participants (inc. tutorial / survey)



presentation


The seminar proceeded well – fortunately, there were few problems of the kind alluded to by Jobs, in the quote starting this chapter. One computer exhibited the font / memory issue, mentioned in 5.2.1 – though this was soon rectified by switching the participant to another computer. Indeed, communal (shared or networked) computers are notorious sources of oft bizarre behaviour in many programs⁷⁵ – invariably deriving from the usurpation of disk space by users. One more serious quirk of these machines was a bug in the  key; *Windows™* only seemed to acknowledge such key presses when twice input. In VSTrack, this generally meant that the pattern editor's *panning set mode* (discussed at various points in the sub-sections of section 4.2) was inaccessible. Fortunately, the demonstration machine (at the head of the room, feeding the projector) had been imported especially for the event, and showed no such problems – allowing this significant feature to be nonetheless illuminated.

fig.5.3 – the author demonstrating the VSTrack application



⁷⁵ The computers used in the seminar had previously suffered issues (unrelated to VSTrack) with numerous applications, including *Cubase*, *Logic*, *Sound Forge* and *Adobe Premiere*, among others.

Furthermore, the propensity for computer users to ignore instruction manuals was demonstrated by the audience's indifference to the provided tutorial – used in stage 2 of the aforementioned schedule. Fortunately (and significantly), most were able to intuit much of the program – and the occasional stumbling block was very easily and quickly explained.

reception

As with the Internet feedback, the VSTrack concept was well received, and individual experiences of the programming were generally favourable. One of the attendees, guitarist Dave Frew – who had previously partnered with the author on a musical commission that had employed the *Renoise* tracker – observed “Though I use the mouse a lot, it makes me use the keyboard much more... you have helped change my approach to making music”, commenting that the interface was “effective and simple to follow”, and that one could obtain “quite dramatic results – quickly”.

perspectives on
computer and
musical interfaces

Among the audience, experience with computers was reasonable – many citing the course itself as their principal exposure. However, compared to the Internet feedback, the various versions of Apple MacOS (9 and X) were much better represented as OS preferences; those with predominantly music backgrounds tended towards MacOS, whereas others favoured Windows™ (notably XP). The musical slant of the audience also influenced input device preference; the MIDI keyboard shared first place with the computer keyboard, relegating the mouse to a distant third – soon followed by more specific preferences, including MIDI controllers and graphics tablets. In the seminar's discussions, all readily conceded the potential for the computer keyboard, in the capacity of musical data entry, notably in the area of laptop usage. However, many also observed that “keyboard commands slow the immediate response as a new user”, and agreed that mouse support would be a vital ingredient in making VSTrack more accessible in the early stages – several also advocating the provision of more “continuous” and graphical control methods.

As expected, sequencer experience was universal – all participants had a reasonable grounding in MIDI. The MMT course exposes its students to several sequencer packages, allowing them to make an informed choice regarding preference. To this end, Cubase's friendlier interface and stability gave it a slight advantage over Logic.

surround sound

The biggest surprise of the seminar, however, was the enthusiasm for surround sound support in future versions of VSTrack – many seeing it as the most likely vessel for trackers into the sequencer market. In this respect, though the tracking method might have to compete with that of sequencing (and MIDI), there exists little in the way of an established standard for surround sound production – and no requirement to supplant any deeply-ingrained conventions or institutions.

Indeed, coupled with improvements in the interface (notably mouse support), the addition of surround sound support led to a majority of people (5/8) who foresaw themselves using VSTrack in the future.

closing remarks

In the end, despite the modest turnout, the seminar was a success – stimulating an enthusiastic discussion and providing a large amount of useful feedback. Many partakers not only appreciated the plug-in in its current form, but enthusiastically pondered the potential of VSTrack in the future. Significantly, the notion of a video tracker (briefly hypothesised in 3.1.4) prompted interest.

Before closing this section, however, some additional, more specific comments from the seminar should be noted, such as: a desire to replace the use of hexadecimal digits with more intuitive decimal readouts; the wish to have a setting determining whether row numbers, samples, instruments, patterns, etc. were numbered from 0 (sympathising with previous trackers) or 1 (empathising with musicians); likewise, the wish to replace tracker terminology (e.g. “song variables”) with conventional computer / sequencer parlance (e.g. “set-up”); a suggestion to standardise some keyboard shortcuts (e.g. “CTRL+V = PASTE rather than ALT+P”); the need for facilities to monitor playback (e.g. an *info page* – as discussed in section 4.2.2); a need to enlarge the interface (pertaining to both the use of larger fonts and the availability of full-screen modes – see section 3.2.2); and both a need for finer resolution, together with a paradoxical appeal relating to the current limits of tracker notation (translating to a “temporal accuracy”).