



Oliver Bown: Creative Technologist

Everyone improves from seeing the reaction to their work, so it's productive to have people around you constantly critiquing what you do...in that sense thriving from collaboration is also very confident and trusting – you are constantly expected to put what you've done in front of people.

Oliver Bown is a music maker and creative technologist with a grounding in several disciplines¹. He is also a researcher who is interested in how artists, designers and musicians can use digital technologies to produce their creative works. In 2015, he took up a post at the University of New South Wales as a senior lecturer and researcher at the Interactive Media Lab. His creative life is collaborative as an electronic musician and programmer for interactive and sound installations. In 2012, Icarus (Ollie Bown and Sam Britton) were commissioned by the electronic music producer Aphex Twin to create the live performance software for his Remote Orchestra project. The concept involved giving an entire 32-piece orchestra headphones over which simple tones could be sent for the players to match and was performed at the Barbican Hall London in 2012. Ollie is the sound artist for *Squidsoup* and works closely with Anthony Rowe and other members of the group in creating powerful immersive interactive installations. Writing software for the sound elements of interactive art installations as well as for music performances is a significant aspect of his creative practice. He sees programming as essentially a conscious thinking activity but one where the intensity of absorption can be so profound that it feels 'effortless' and his awareness of time itself disappears. This 'extraordinary feeling' is one that is familiar to many creative practitioners working on very different activities: the visual artist whose drawing makes her unaware of the people around her or the musician improvising with others across the Internet who enters a state that feels 'ethereal'.

An interview follows in which Ollie talks about his creative practice in collaboration.

Q: How would you describe the work you do?

O: I am very spread in my interests - a jack of all trades. One component is an electronic music practitioner, another is a creative technology practitioner in creative coding and software development for music and installations and so on. Another part is computational creativity research which involves some software development and experimental research. But also, the theory around it so that goes full circle in terms of the reflective practice because I spend a lot of time thinking and reading about creativity. My training is also a mixed bag. I studied mathematics and social anthropology one after the other in my undergrad. Then evolutionary and adaptive systems at Sussex University which was on artificial life and cognitive science as well. Then a PhD at Goldsmiths where I applied those evolutionary models to human-musical behaviour and looked at cultural evolutionary processes. Then I worked with Jon McCormack in Melbourne in generative, evolutionary art. Then I came to the Design Lab (Sydney University) where I drew in more design focus work. I'm now at the Interactive Media Lab at UNSW's Faculty of Art and Design. My music practice has carried on throughout through different ensembles, mainly the improvising group Tangents.

Q: When you are engaged in writing software when is the most intense, focused work?

O: Certainly programming, of all the activities I have ever engaged in in my life, is by far the most absorbing and immersive experience. That concept of 'Flow' is immediately recognisable when you're there and you step up and you haven't eaten for four hours, you've completely lost track of time and your entire brain and body is focused on a problem you are working on and it's an extraordinary feeling.

Q: Some people say 'when I am doing it, I don't think'. Do you have a sense of that when you are programming?

O: It's hard to describe programming as an activity where you don't think- you are so obviously thinking in various ways- and quite consciously. But what is interesting is when it becomes so effortless- in a sense where you don't get distracted, you are not constantly going on Facebook or making cups of tea, you are actually

locked into it so it becomes this passive thinking in a way. You are aware of the fact that you are solving problems but the flow is so natural that there is no struggle to do it.

I think it is exactly as Csikszentmihalyi describes it: if you are working on a problem that's too difficult you get frustrated and give up; if it is too repetitive and easy, you find it banal and boring but there is that sweet spot where there's a certain challenge- usually it's something you've learnt how to do before but you are still not very good at it. It's the most seductive, addictive challenge.

Q: Is there a moment when you might pause in that process and the thinking becomes more like reflection?

O: There's a couple of different levels of reflection and one is ... Well I am not always doing this in a creative context because when you are making a library for something you are making it more as a software engineer/designer and you've got quite clear objective goals... Sometimes I am developing those core libraries in that mode and sometimes I am making creative work. So starting with the latter- the creative practice- there's the reflection on what you are doing aesthetically and the having to change hats from a programmer to artist-playing split personality.

Q: When does that happen in the process?

O: It usually happens in bursts because usually you have a short task that you need to achieve- that requires me to write these libraries /functions- so that will be half an hour of work before your head comes up again and you stop to look at what you've just done... I would say that one typical mode is that's usually in half hour to an hour blocks, where you are developing something and then stopping and you've got this chance to listen. It's a very slow process. It's not the sculptor stepping back looking at the clay, it's a much longer form. Then there's the much more rapid turnover mode where all you're doing is getting down to the point of adjusting some parameters. And so, you've set up a program that runs some generative process and you're tweaking it; you are just literally trying to find the right numbers, the parameters in the system. That's happening a lot more rapidly.

The vision of the Max/MSP people was that you are coding whilst listening: you are locked into this constant iterative cycle which is very rapid and effective, but that will always potentially fall apart because you have to go and engage in some other task that's going to take you half an hour because you realise this is not working and you need to fix something.

Q: Last year I was talking with Ben Carey who was saying because he plays an instrument with his software that he does quite a lot of tweaking of the software in response to the performance element. Is there any of that in what you do?

O: In the sense that I'll be performing but I'll be performing on the laptop- controlling the software that I am making. So, there's the sense of running it, trying it, having a play, seeing how it feels so that brings in questions like what the right interface elements are... Aesthetic questions but also practical questions like 'if I press this button, the sound explodes or the program crashes. As a performer I can choose to make sure I don't press that button or I can go and fix the software.

I work a lot with granular synthesis². When you are setting up the granulation of a sound, you'll be listening to that and adjusting parameters to get it to sound smooth or to get it to introduce interesting modulations on an effect. Rhythmically I do a lot of stuff with stochastic chopping up of beat patterns; so it's just finding the right numbers that work or the right probability ratios that work to get what sounds like a musically coherent pattern. Trying to find things that change nicely over time. Never go into bad areas- so trying to define the space where the system is searching, constantly varying but never going bad because you don't want that to happen in a performance.

Q: So you develop your own musical work, the software and then you become part of a collaboration- whether you are working with Squidsoup or somebody else, how does that affect what you do? Do you need your own brief, freedom of movement?

O: For most of the stuff I've done with Squidsoup I've been the sound guy and the rest of the team has worked on the light element and that is a nice clear distinction. Ant (AP) has me on the team in the sense that he is fully confident that whatever I do he's happy with aesthetically and technically. He just needs to tell me what the lights are going to do and I'll go off and make something that works with them.

Q: How does that work?

O: That's a good question because it's not that straightforward. Generally, the vision of these things is that they are tightly integrated audio-visual experiences: those two elements are worked on together and tightly integrated and in practical terms that means you've got one thing that's running the process and driving the other thing. A lot of this comes down to the social and practical technicalities; who's on the team. How much time have we got? What software is already made? All of those things will influence what path you take so the most common iteration is that he and his light team write a program that controls the lights in some generative or interactive way and then broadcasts state information about what's going on with the lights. Then I plug in a separate computer which is receiving those messages and I write a computer music program that's being driven off those messages. A banal example: there's an explosion that happens in terms of a growing sphere of light and that explosion will be signified by a single message that says <explosion here> with some parameters- light, intensity, locations and so on. So then I can choose which of those parameters to use. That's a neat separation because I can sit there in my room coding that stuff up and they can sit there and we just have to communicate the messaging protocol that we'll be using to connect the two elements.

That works for the vast majority of situations but there's a few problems: one is you do have to get together and see what you've got at some point. I can make the system but if my expectation about what messages are coming in is wrong then it's not going to work. Even if I do have the right idea we still see something that we've never seen before when we put those things together and we'll obviously reflect at that moment.

With *Squidsoup* most of that work is done in separate places which makes the whole iteration really interesting because you've got this whole development happening blind... And then there's this established format now which is that whenever a *Squidsoup* piece goes up, it takes a day to put the thing up and the normally there would be one or two days between putting the thing up and it going live, where literally everyone is in the same space with computers, with the real thing, tweaking. It's a very limited amount of time but that way we know that the thing we are making is the thing people will see, rather than doing all that guess work. So there's a very interesting pathway towards the moment when you've got the true reflection of what you're making and then all these anticipated reflections....

Q: How often do you get what Schön called the 'undesirable surprises'? where you've made some assumptions and something happens and it's not what you want.

O: Pretty often, that's pretty common.

Q: What do they (the surprises) sound and look like?

O: Usually flat, disappointing in the sense of ...I am thinking of an example- the week we spent in Amsterdam creating this piece and that was an inversion where I wrote the software with the audio that drove light: so the lights were responding to what I was doing – an interesting inversion of the whole thing. So, a week to run something was quite a long time was high pressure and to do it all in code rather than to be able to drag and drop and listen. In that case, you would spend.... I work in a very modular way so that means I've got an idea for an instrument process that I'm interested in and I'll make a module that does that and then the idea is that the module would then be re-configured in various ways.

O: In object-oriented programming that's like making a base class and then making lots of children classes of that. The way that my search process would be to work out what the design of this thing was going to be –this abstraction- and from that spawn a whole load of processes. In order to work out what the thing is going to be you have to have a couple of prototype instances happening so you would be developing those in parallel, moving back and forth between the instance and the base class until you were happy with how the base class was working and then you'd spin off four or five variations on that. There is a huge leap to get to the base class thing and nothing has really happened- you haven't managed to make anything. But then very rapidly you get to create these variations. That's the point where you get to hear what you're doing and reflect. Often, it's not what you want, but at this point you've committed to a certain design and it's often the case that rather than going back to the drawing board you reconsider what your goals are, in a Schönian way, reconceptualise the problem. In that sense it's almost never the case that the base class is *wrong* because you can always reconfigure and rethink what you are doing. The idea is always fine you just have to find that configuration that makes the idea work. Of course, the idea may not be quite as exciting as you thought it would be but it would be rare that you would have to completely scrap that. You'd keep going nevertheless.

O: So there's the slow build up to that point, a slow iteration between the base class and the instances as you work out what you want the base class to be like and then a very rapid development of instances and during that you are more likely to come across stuff that sounds horrendous and say 'I'll bin that' and keep trying. It's more like a brainstorm at that point.

The Processing approach to creative coding is about sketching, you are doing your rapid broad search early on and probably nothing more because generally the idea is to make simple programs. My approach is you build your platform which you are going to explore and then you can start to do your broad search but that comes quite a lot later. That's what I like to do for those Squidsoup moments where we all get together in the room at the very last minute. Because it's the very last minute you want to have as much of your toolkit ready to go. Then you are doing this much more banal direct 'let's just drop this sound in here' etc. It becomes more like a traditional 'composerly' activity in a very constrained space.

Q: When you are in the same space, when you are all together does it work because of particular factors such as having a shared aesthetic of some kind?

O: I think it works because there's a neat distinction between the lights guys and me doing the sound. Because there's enough mutual admiration for what each other's doing that's one of the foundations and you can't really go wrong once you've got to that point. In addition, there's a good understanding of what each other's doing so there's no surprises. You are aware of the tropes within which everybody's working and once in a while it will get very exciting because you will move beyond those tropes and do something no one has seen before.

I am just trying to come to terms with what you mean by a 'shared' aesthetic. There is an 'agreed' aesthetic which is not always perfectly defined but at the moment all the discussions have been had it's all go. The classic example with Ant's work: he wouldn't do anything symbolic or social or political, and he also is quite keen not to do anything representational. He positions himself as a light artist or an architectural media artist so even having an image of a person displayed in the space, which would be so easy to do- to represent a body or face in that context is a "no, no" as far as Ant's concerned ... Everyone is on board with that- they see the vision. They understand.

Related to that discussion about aesthetics is our having shared heroes: Ant is very into artists like Ryoji Ikeda and that minimalism. I am to a lesser extent. We have other influences musically but he has known my music for a very long time so he understands what I do. I don't think it hugely influences the way he goes about doing the work: I do think we live in a 'visuals first' world except when a live band is playing and the VJ is following them, but generally the sound is the thing in collaboration that falls into place last.

Q: How long have you been working with Squidsoup?

O: I have been working with them for ages. I think I must have met Ant in 2003 and originally as an electronic musician just doing sound things. He used to be in Shoreditch with a little web development firm and that's what Squidsoup was doing, commercial projects – I think we did the Lichtenstein National lottery at one stage. At that point I wasn't even a programmer. It has been a long time coming that I have actually worked with Squidsoup on the programming front.

For a very long time, even with the *Ocean of Light* stuff we were still basically making fixed audio-soundtracks. That was quite frustrating, having trained myself up to become an interactive audio programmer but there was never the time. I had the time but that cost of getting those messages correctly formatted and sent around and also the risk of running computers in real time and me being on the other side of the world. I think those practical things end up becoming really interesting because they shape so much work.

Q: How do you handle this kind of work from a distance? Do you need to see people to talk about it or can you do it by just swapping files?

O: I find it varies massively. Every detail can be a confounder for the potential of that collaboration: examples include working with my cousin, in my duo Icarus: he's based in the UK and we've worked together for ever so we have an incredibly close understanding about what we are doing- we've worked on 10 albums together. One of the reasons we are not working so successfully at the moment is that he runs a studio and so works with a load of expensive plugins in Ableton and we are sharing projects over the Internet and I don't necessarily have those same plugins. I can't hear the projects the way he's hearing them. There are many workarounds to this, but it just ever so slightly adds drag to the collaborative effort and that can be make or break to how well the collaboration

excels. That's one example. One of the domains where it's more successful is in software development and you've got incredible support tools and everything's just text that takes no disk space...so it's easy to have a shared development environment and keep the flow.

The time-zone one is a huge thing. You can pick up the phone and chat when you are working on something but not if the person is asleep in the middle of the night. Sometimes you have a question and it can't be answered for the next twelve hours and that can ruin your flow.

Then there are inter-personal issues and issues of authorship. Sometimes people get very grumpy with someone else, fighting over what they're doing. I've seen that in various ways- getting upset myself or someone getting upset with something I've done; strong personalities trying to steer the ship. That's dysfunctional even when you are together but it can become really confounded over a distance. For example, working together just on a paper, someone's style might be to go in and aggressively edit with the expectation that everyone's happy with that, someone else might be more defensive- you can't touch that because I wrote it. So, there's strong interpersonal factors. I think there is no correct style or approach, it's all about trust and mutual experience. I work with very different people with very different styles. It's great when people are accommodating, but you also want to work with people who are passionate enough about what they're doing that you might have to have 'robust' disagreements.

Q: Do you have a track changes method? Or do you just look at different versions?

O: Those tools can be good but they can also break the flow: trying to read something in track changes can be a nightmare. But those things can work really well in the world of text. They work brilliantly in the world of code if you get to know github. (a website that hosts projects that are shared using the "git" technology. It is the most advanced collaborative technology I've ever used and used by programmers the world over). If you get really good at them they can be supremely powerful because if you check in online and you've got this detailed history of who did what, added comments to say this is why I did this blah blah... There's no way to work in an everyday program like Ableton Live where I am able not just to see the track but all the comments of why I did this and why I did that...Is it complete or pending or? Those minor details make a huge difference. (Post-script comment -, Dropbox has added a load of features that allow you to comment on file versions and maintain a collaborative conversation as a kind of meta-layer over the files you're working with. I haven't got into the habit of using those yet. One thing I have got into is Slack, which is very fluid and effective and integrates well with Google Drive, but I haven't seen that work well yet with large files and media projects).

Q: How often do you actually look at the record? It must take time to go back and check everything.

O: There is time-lag. How often depends on the level of intensity. I am working a record with people in Sydney at the moment but we are still doing everything in Dropbox. I'm doing most of the production so I'm rendering - not everyone has Ableton and can listen to the material - so I am rendering out roughs every time I make an edit. Every time I save the file I also export the .wav file of the track. In theory everyone's got that on their phone in the street or driving their car and they can just check out the latest. That's incredibly productive. For me with music production having lots of time to just listen in your car, listen on your headphones or listen in your kitchen while you're cooking dinner, is the best way to get the music good. The fact that you can do a rapid deployment to all of your team is really good.

Going back to something like Squidsoup's installation work, that's hopelessly impossible. And that's one of the things I am very keen to work in my research towards is creating amazing simulation tools that would allow you to...I mean why not be able to throw on your VR goggles and Ant can show me the latest imagery that he's created and I can play him the latest sound and that should just be as quick.

Q: Would it be the same as entering the physical space?

O: No. Ant is very adamant about that: the simulation is not going to be the same but I take the view that yes that's true but having seen the real thing and then going back to the simulation, you know what's wrong with it, you can fill in the gaps, or you can interpret...The important thing is- say I am interested in how this thing works with these blobs moving across the screen, it doesn't matter that I'm not there.

Q: I think it was the installation in Salford Quays- it was 'Submergence'. Anthony said that the only way you can understand it is by getting inside it... with works like this that come right out into the public space, when you experience it for yourself, do you find that the experience is surprising or unexpected in any way?

O: I thought the key thing about being able to go *inside* the *Ocean of Light* system is that if you start to make shapes and things that are all coherent and make a lot of sense when you are looking from outside and at a distance, the further away the higher resolution it is and the more it makes sense as a structure. When you're standing inside it's like putting your eyes right up to the TV screen, you completely lose stuff. So there is this hierarchy of what becomes perceptible when you are inside, shapes that are right up close to you become imperceptible from what they are supposed to be but movements still work so something moving up from the ground up, it doesn't matter that you are so close to it, you can still sense that. You can list the things that work from outside and the things that suddenly fall apart when you go inside. And that's the golden rule. You can write that down as a written rule that anyone making stuff beware, when you go inside this thing it's not going to look like that. Ant told me that and I still underestimated how true that was. I was still imagining certain things would work like say a cuboid appearing around you would make you feel like you were in a box and it's not true because it doesn't work at that distance.

Q: Why has Squidsoup worked so well and for so long?

O: Ant (Anthony Rowe) is the hub and the point of contact for everyone. It's a completely geographically distributed group. It is a very interesting group to be part of. Maybe it's better to describe it as just Ant and he farms out jobs to people and gets them involved. So, if you think of it- he could have positioned himself as a media artist under his name rather than giving it a name like Squidsoup and making it a collective. He could have done it that way, like many artists do, and of course all of those people have people working for them who are contributing elements. He is much more comfortable working in a group and is very consultative and inclusive. I think he likes to co-develop an idea, to work something up with somebody.

Q: Are you the same?

O: I am the same. Musically - and this might be because I'm not a trained musician with an instrumental background, so I'm not quite as effective as I'd like to be working solo. But for one reason or another I've never worked solo as a musician. I've always collaborated and for a long time craved to do something solo and really mark out a name for myself in that way and to have to come to terms with the fact that that's not for me. Partly that's because if I work with really good musicians, not acting as session musicians but actually working with them interactively, then that brings out the best in me. But also, it's a bit of an emotional thing, it's that thing of maintaining motivation and the pressure of producing; and sometimes I think that just huddling together in a group is more comforting... something falls over when I am engaged in that process which doesn't fall over when I'm working in a group. Everyone improves from seeing the reaction to their work, so it's productive to have people around you constantly critiquing what you do. In that sense thriving from collaboration is also a very confident and trusting stance - you are constantly expected to put what you've done in front of people. That idea of bouncing ideas off other people or working up ideas or using other people to engage in that iterative process that you can do in your own head but you don't- you get other heads involved... I worked with Ben Carey last year on an album and at some point, I quipped that "I feel so much smarter having two brains." That is what it feels like sometimes.

¹ <https://artdesign.unsw.edu.au/about-us/our-staff/dr-oliver-bown> and <http://www.olliebown.com>

² a method by which sounds are broken into tiny grains which are then redistributed and reorganised to form other sounds. <http://www.soundonsound.com/techniques/granular-synthesis>