Music with Unconventional Computing: A System for Physarum Polycephalum Sound Synthesis

Edward Braund and Eduardo Miranda

Interdisciplinary Centre for Computer Music Research (ICCMR)
Plymouth University, UK
edward.braund@students.plymouth.ac.uk
eduardo.miranda@plymouth.ac.uk

Abstract. The field of computer music is evolving in tandem with advances in computer science. Our research is interested in how the developing field of unconventional computation may provide new pathways for music and music technologies. In this paper we present the development of a system for harnessing the biological computing substrate *Physarum Polycephalum* for sonification. *Physarum Polycephalum* is a large single cell with a myriad of diploid nuclei, which moves like a giant amoeba in its pursuit for food. The organism is amorphous, and although without a brain or any serving centre of control, can respond to the environmental conditions that surround it.

Keywords: Music with Unconventional Computing, *Physarum Polycephalum*, Sonification, Bionic Engineering, Unconventional Computing

1 Introduction

Since its invention, the computer has become increasingly ubiquitous in everyday life. In music it offers a seemingly limitless paradigm for composition and consumption. Historically, one of the first known computer music implementations was during the 1950s. Here, the musical curiosity of a computer scientist enticed him to programme the CSIR Mk1 machine to play a selection of popular melodies [9]. Shortly after his experiments, composers and scientists began exploring the computer's ability to be a tool in the composition of music. One of the first pieces to contain computer generated material was the *Illiac Suite for String Quartet* [10], which was written by a mathematician and composer in the late 1950s. Since these early computer music experimentations, advances in computer science and music/sound have demonstrated a close correlation. Consequently, computational advancements have impacted instrumentation and compositional practices resulting in novel genres of music as well as new methods of distribution [15, 16]. There are genres of music today which consist mainly of computer generated sounds. It is therefore expected that future developments in computer science will continue to have a strong influence on the field of music and related technologies.

We are interested in exploring ways in which the advancing field of unconventional computation may offer new pathways for computer music. Unconventional computing is a branch of computer science that addresses computing paradigms other than the conventional Von Neumann architecture and Turing machine, which have dominated computing since the 1930s. These non-standard models look to the information processing abilities of biological, chemical and physical systems, and how they may be exploited as either a genuine or utopian computational model. In this paper we report on the initial results of our work into harnessing the biological organism *Physarum polycephalum*, henceforth known as *P.polycephalum*, for use in the field of computer music. Specifically, we present the development of a system for recording the behaviour of *P.polycephalum* for sonification. At this early stage of our research, we are focusing on how the technique of sonification can be used to begin understanding the behaviour of this emerging computing substrate. We anticipate that this will provide us with an indication of which direction to take our research going forward.

This paper is structured as follows. First, some background information on the project is explained, offering knowledge on unconventional computing, some details on other projects in the area and an introduction to *P.polycephalum*. Next, the development of our system for collecting behavioural information from *P.polycephalum* is presented, followed by its testing. Then, the implementation of a simple sonification experiment is explained. Finally, the paper ends with final remarks.

2 Project Background

To cite this research let us first look to the development of today's conventional computer, which lays its ancestral roots with the Turing Machine pioneered in the 1930s [21]. This abstract computational model was developed by formalising the behaviour of 'real world' computers: large groups of people who carried out calculations following a strict procedure [8]. Shortly after this invention, Von Neumann developed a stored-program computing architecture [7], which widened the scope of the computing machine. These inventions saw the deterioration of 'real world' computers and, through the course of 80 years, have become the father of today's conventional computer. Over this period the premise of computation has remained relatively unchanged, with Turing's idea of a computing machine leading to the ideology of a universal computer that could solve most mathematical or logic based problems. This left non-standard computing models mainly residing in the theoretical domain until recently where momentum has been building due to a growing need for faster and more efficient technology.

Unconventional computing models currently being developed harness abstractions from a wide selection of phenomena. These range from reaction-diffusion [4] to emerging quantum computers [11]. Such computing devices, amongst several others, are exciting scientists who speculate that if this area of computer science is developed at the same frequency as the Turing Machine then *"Seventy years from now, the technology will be unrecognisable from today's ideas"* [19].

In regards to music, we consider there to be two approaches to harnessing unconventional computing. We refer to these as the algorithmic and sonic approaches. The algorithmic approach relates to how unconventional techniques are harnessed within other disciplines. For instance, an algorithmic implementation could produce the arrangement of musical sections or create an environment for working and interacting with music. The sonic approach on the other hand is uniquely attributed to unconventional computation in computer music. Here the computational behaviour of the harnessed phenomena is exploited to produce sound in a sonification model. Its behaviour can then be controlled to alter the model's parameters, allowing variations in sonic material to be produced. This approach lends itself to the development of new instrumentation.

The technique of sonification can be described as the art of using non-speech audio to convey information, and has a truly interdisciplinary background with it being employed for purposes ranging from recreational to scientific. Sonification is an elementary starting point for harnessing unconventional computation in computer music. This is because it allows for knowledge of natural systems to be realised and relationships to be drawn between behaviours and music/sound. The outcome of which could indicate possible computer music applications that the phenomena may be suited to.

Although research into unconventional computing in music is in its infancy, there are a number of projects beginning to emerge. To the extent of our knowledge, these mainly adopt a sonic approach. One early example explored using chemical computing by way of a Cellular Automata model to control a granular synthesiser [14]. Another example investigated synthesising sounds with a hybrid wetware-silicon device using in vitro neuronal networks [13].

Regarding our research, there are many unconventional computing prototypes currently being developed that could hold potential for computer music. However, many of these require expensive laboratory equipment along with specialist knowledge to allow computational prototypes to be developed. At this stage of our research we needed a more accessible medium to begin conducting our experiments. Uniquely, the biological substance *P.polycephalum* requires comparatively less resources than most other unconventional computing substrates: the organism is cheap, openly obtainable, considered safe to use and has a robustness that allows for ease of application. Moreover, *P.polycephalum* has been building presence in unconventional computation studies with evolving notions of it being a *"universal computer"* [1]. For these reasons, we have selected *P.polycephalum* as the computing substrate for our research.

3 *P.Polycephalum*

P.polycephalum is an acellular slime mould belonging to the order *Physarales*, subclass *Myxogastromycetidae*, class *Myxomycete*. In the field, this organism exhibits a dynamic life cycle of thirteen phases, which sees it develop from spore germination to its main vegetative plasmodium state. It is *P.polycephalum's* plasmodium state that yields interest for unconventional computation.

As plasmodium, *P.polycephalum* exists as a large single cell (visible to the unaided eye) with a myriad of diploid nuclei, which divide in natural synchrony through mitosis every ten or so hours. The organism displays negative phototaxis and subsequently resides in dark, cool and damp environments. Its appearance is a yellow mass of protoplasm that moves like a giant amoeba in its pursuit for food. Plasmodium propagates towards chemo-attractants and away from chemo-repellents, which have formed a gradient on a substrate. Propagation is achieved by extending pseudopods, which disperse forming a search front while building a route-efficient network of protoplasmic veins connecting foraging efforts and areas of colonisation (Fig 1). Upon discovery of food, the plasmodium surrounds it with pseudopods and feeds through the process of phagocytosis, ingesting nutrients that are spread across the organism via cytoplasmic streaming. Conversely, if matter is discovered which does not entice the appetite of the plasmodium, the area is avoided. Over time, given the correct environmental conditions, the organism can grow to become a considerable size and propagate at speeds of up to 5cm/h. The visual result of a fully-grown culture is a planar graph with nodes represented by areas of colonization and edges by protoplasmic veins, as shown in Fig 2.

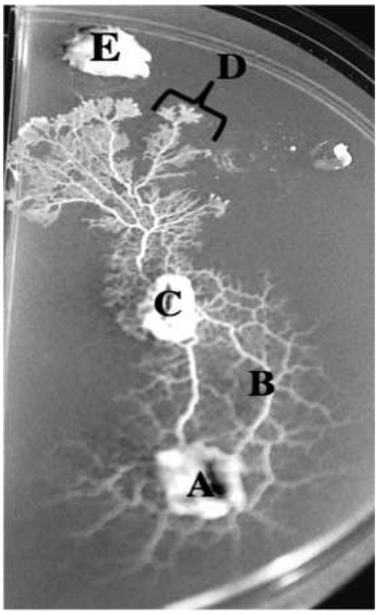


Fig. 1. A photograph of plasmodium of *P.polycephalum* showing: (A) inoculation of plasmodium into the environment, (B) protoplasmic network connecting areas of colonisation, (C) colonised food sources, and (D) extending pseudopods forming a search front along a gradient to food marked by (E).

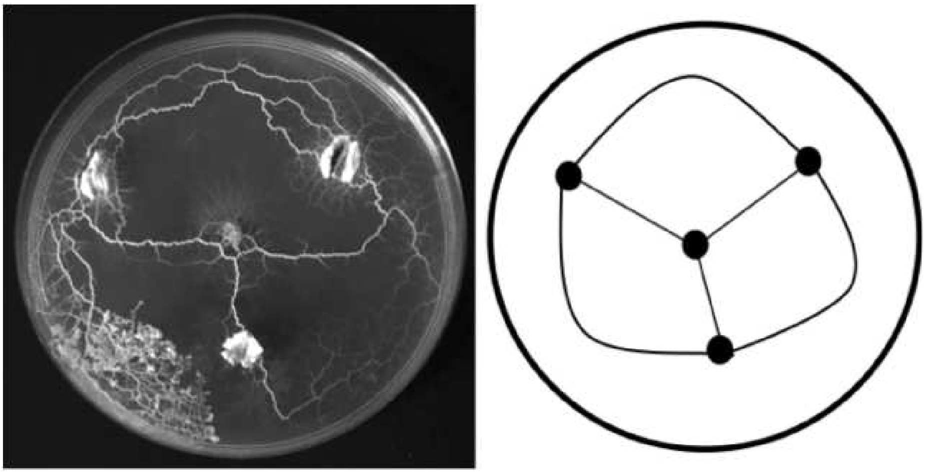


Fig. 2. The visual result of a culture of plasmodium in relation to a planar graph. Shown are several areas of colonisation (nodes) connected via a network of protoplasmic veins (edges).

The intracellular topology of plasmodium has been described as a network of biochemical oscillators [18]: waves of contraction or relaxation which collide inducing cytoplasmic streaming. This intracellular activity produces fluctuating levels of electrical potential, typically in the range of ±50mV, displaying oscillations at periods of approximately 50-200 seconds with an amplitude of 5-10mV [12]. If recorded in isolated zones of colonisation over the duration of it being active, patterns emerge that correlate to spatial activity and environmental conditions. Adamatzky and Jones have examined such patterns and have reported that they can be used to denote the plasmodium's behaviour and physiological state [6].

P.polycephalum has been the subject of advancing research in recent years due to the sophisticated behaviour it demonstrates while foraging: the organism reacts to environmental stimuli with parallelism; allowing it to find efficient routes to food, build an optimised network of protoplasmic veins, and avoid unfavourable environmental factors. In computer science, researchers have explored using the organism as a parallel computing device capable of solving a wide selection of problems. The genesis of computing with *P.polycephalum* was a maze solving experiment [17], which developed methods of manipulating its chemotaxis to produce a route-efficient protoplasmic vein network from the entrance to the exit. Since this early experiment, use of the organism has advanced to more complex applications such as robot control [20]. In this project scientists developed a six-legged robot whose movement was controlled remotely by *P.polycephalum*. The robot was fitted with light sensors that relayed intensity information to a computer. This information was then used to project white light onto select areas of the *P.polycephalum* culture. The resulting phototaxis was recorded and used to naturally manoeuvre the robot away from light. Other laboratory proofs of *P.polycephalum's* computational ability include route planning [2,5], colour sensing [3] and numerous others [1].

4 System for Recording the Behaviour of *P.polycephalum*

In this section, our system for recording the behaviour of *P.polycephalum* is presented. To ensure we produced a system fit for purpose, two criteria focused the development process. The first was to select methods of recording behaviour that were discrete, ensuring they had no effect on *P.polycephalum.* The second was to construct a system whose output could be adaptable to accommodate different types of future experimentation.

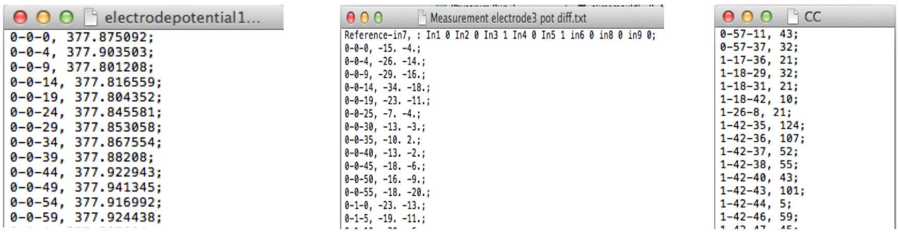
As the organism takes several days to exhibit substantial growth, the possibility of harnessing it in real-time is limited. As a result, *P.polycephalum* computational devices are commonly recorded using time-lapse imagery. This allows for behaviour to be interpreted via human interface, or complex image-tracking algorithms, both of which are not efficient or guaranteed to be accurate. For sonification, we needed to develop more tangible methods of gathering detailed information of its behaviour. To achieve this, we looked to the research discussed previously regarding *P.polycephalum's* electrical activities [6]. This method of recording behaviour provides a high level of detail regarding the organism's evolving physiological states, in a form that is very usable. Recording behaviour this way requires electrodes to be positioned at points of colonisation and isolated from one another using a non-conductive material. As a single entity, this, unfortunately, would not allow the organism's propagation trajectories between electrode zones to be recorded. To combat this, we proposed the use of both time-lapse imagery and electrical potentials to record behaviour, as it provides information on stationary colonisation, as well as propagation trajectories.

4.1 The System

To enable us to capture *P.polycephalum's* behaviour, two forms of hardware are required: a camera with a flash and an ADC interface with a high resolution. The camera we selected was a USB high definition microscope with a manual focus. This device has eight white light LEDs that can be used as a flash to illuminate the foraging environment appropriately: it is important that the environment is only illuminated momentarily for image capture to avoid affecting *P.polycephalum's* phototaxis. To record the electrical activity, a 20-bit ADC high-resolution interface manufactured by Pico Technology UK was selected. This device can facilitate up to eight single-ended electrodes. When using this interface to record *P.polycephalum*, electrode arrays have be arranged with one reference and a number of measurement electrodes.

To facilitate the collection and collation of information from these hardware devices, we designed a piece of bespoke software suitable for our research. We programmed this in Max, by Cycling 74, as it offers a comprehensive tool kit for data acquisition, as well as continuing operations from the collation stage.

We developed our software to offer two collation frameworks for sampling electrical activity: Electrical Potential Collation (EPC) and Custom Collation (CC). EPC operates one systematic collation protocol for all inputs, at definable intervals. The CC framework consists of two selectable collation protocols: Electrical Potential Difference collation (EPD) and Conditional Collation Gate (CCG). EPD calculates the potential difference between a selected reference input and a selection of other active inputs. This protocol collates at definable intervals and includes a reference line at the beginning of each collation to denote its contents: a line of text that states the name of the reference source and each selected input. Such a collation is particularly useful as it compares the level of activity across an environment. CCG differs from the previous two protocols as it does not conform to set intervals. The protocol collates readings from each input at the exact time one of them reaches a definable increment/decrement threshold from their previous stored reading. This acts as a method of compressing incoming data to retain only changes at a threshold that is deemed necessary. To allow for data to be viewed visually, the contents of each collation is plotted in a live graph.



```
0-0-0, 377.875092;
0-0-4, 377.903503;
0-0-9, 377.801208;
0-0-14, 377.816559;
0-0-19, 377.804352;
0-0-24, 377.845581;
0-0-29, 377.853058;
0-0-34, 377.867554;
0-0-39, 377.88208;
0-0-44, 377.922943;
0-0-49, 377.941345;
0-0-54, 377.916992;
0-0-59, 377.924438;
```

```
Reference-in7, : In1 0 In2 0 In3 1 In4 0 In5 1 in6 0 in8 0 in9 0;
0-0-0, -15. -4.;
0-0-4, -26. -14.;
0-0-9, -29. -16.;
0-0-14, -34. -18.;
0-0-19, -23. -11.;
0-0-25, -7. -4.;
0-0-30, -13. -3.;
0-0-35, -10. 2.;
0-0-40, -13. -2.;
0-0-45, -18. -6.;
0-0-50, -16. -9.;
0-0-55, -18. -20.;
0-1-0, -23. -13.;
0-1-5, -19. -11.;
```

```
0-57-11, 43;
0-57-37, 32;
1-17-36, 21;
1-18-29, 32;
1-18-31, 21;
1-18-42, 10;
1-26-8, 21;
1-42-35, 124;
1-42-36, 107;
1-42-37, 52;
1-42-38, 55;
1-42-40, 43;
1-42-43, 101;
1-42-44, 5;
1-42-46, 59;
```

Fig. 3. Collation files created by each of the system's collation protocols. From left to right: EPC, EPD and CCG.

The software's image collation framework operates either at independent intervals or in synchrony with EPC. In addition to this, another collation protocol can be enabled that collates in tandem with CCG, allowing for points of heightened activity to be visually recorded. All collated image files are stored in a predefined directory and can be compiled to produce either a 15fps or 30fps video file.

All the collected data is organised using a global time-elapsed indexing system, which consists of three selectable formats: hours-minutes-seconds, seconds, and reading number. Furthermore, real world time and date information is embedded into each collated image, allowing for changes in behaviours due to daylight or other time-based factors to be recorded. Fig 4 shows the software's user interface, with annotations denoting the function of each panel.

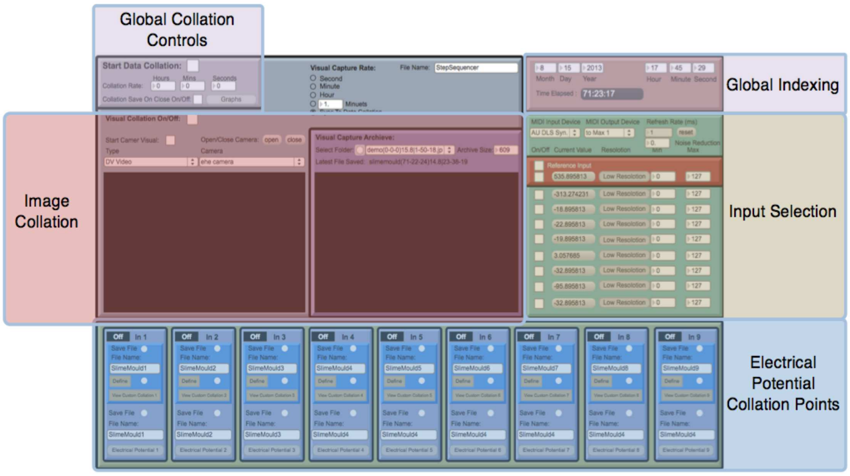


Fig. 4. Collation system's user interface with annotations.

In addition to these operations, the software also offers two methods of post-collation data compression. These are related to electrical activity readings and are included as they will further expand the system's usability. The first compression algorithm groups the collation into blocks containing a defined quantity of readings. It then combines each input's readings, leaving a single arithmetic mean value for each block. This is expressed in the following where $e=$ electrical reading, $r=$ readings in a single block and $a=$ input number:

$$\frac{1}{r}\sum_{i=1}^{r} e_i^a \tag{1}$$

The second compression algorithm views the data not as individual electrode readings but as sets of entries at each collation interval. It then compresses the data set as follows: n entry is only retained if m number of measurements from k amount of readings presents a change over b threshold from the previous entry stored (first entries are always retained). This is expressed in the following where if $x(\mathrm{C})= 1$ the entry is withheld, otherwise it is lost:

$$\sum_{i=1}^{k} x(|a_i^n - a_i^{n-1}| \geq b) \geq m \tag{2}$$

4.2 System Testing

To ensure the operational success of our system, we initially tested the hardware and software in a basic scenario. Here, two electrodes coated in non-nutrient agar were arranged in a linear fashion, with the left elected as reference and inoculated with *P.polycephalum* and the right furnished with an attractant. The system was then initialised to enable all collation systems relating to the single measurement electrode. Fig 5 depicts the hardware setup and electrode arrangement.

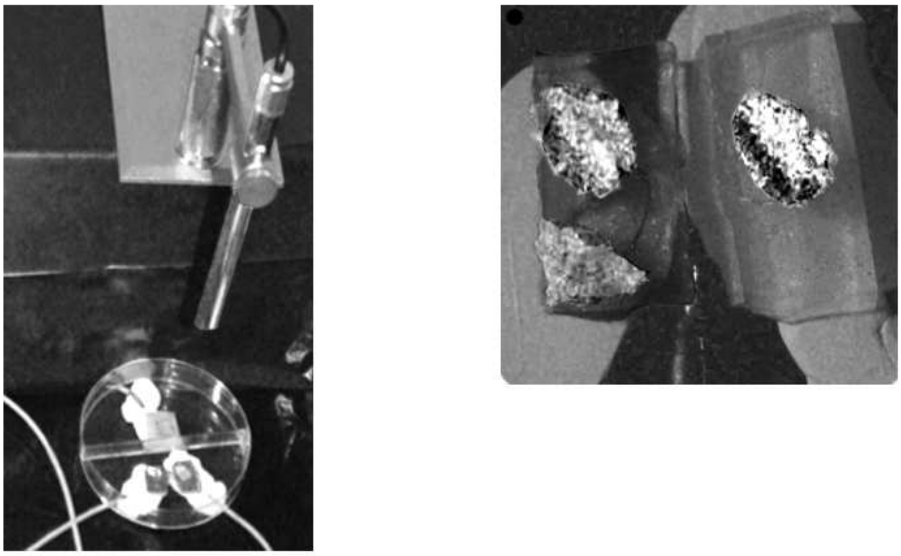


Fig. 5. Photographs of the experimental step up used to test the our system.

This testing experiment was left to run over the course of three days, and was only halted when no change in electrical potential had been exhibited for a period of time. The resulting collation was then scrutinised to ensure it accurately represented behaviour, and that the appropriate correlation existed between recorded activities in time-lapse imagery and electrical potential. Furthermore, the collations were compared to research put forward by Adamatzky and Jones [6] on the connotation of the electrical behaviour of *P.polycephalum*. This reinforced that the collected data was accurate and in line with accepted research within this area. Fig 6 shows this comparison and confirms the system's operations and output were a success.

5 Experimental Sonification

We are currently experimenting with a number of applications that harness behaviour recorded by the system presented in this paper. One of these is briefly explained in this section: the *P.polycephalum* step sequencer. This application is one of our more novel approaches to sonifying behaviour, taking the form of a compositional tool that allows a user to define certain parameters of the sonification model.

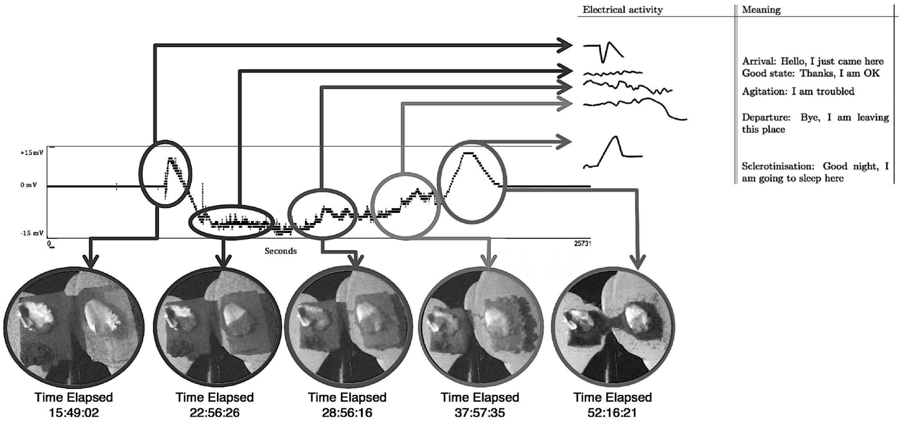


Fig. 6. Correlations between activity recorded in electrical readings and images. Shown is a connotation table courtesy of Adamatzky and Jones [6], which denotes the meaning of electrical patterns.

Step sequencers are devices that loop through a defined number of steps at predetermined time intervals. Normally, each step can exist in either an active or inactive state, which regulates whether a sound event is triggered when the sequencer reaches its respective position in the loop. The idea of a *P.polycephalum* step sequencer was conceived when reviewing sets of time-lapse images with correlating electrical activities. We noticed how the substance oscillates protoplasm around a network of veins to colonised regions, and how this relates to the architecture of a musical step sequencer. Resulting from this thought, we conceptualised a sequencer where *P.polycephalum* controlled step activation through propagation trajectories/colonisation, and sound event triggering with fluctuating levels of electrical activity.

To implement this experiment, we first developed a growth environment that could represent a step sequencer. Here, we arranged six electrodes in a circular fashion with a reference electrode placed in the centre (Fig 7). Each electrode was coated in a non-nutrient agar with an attractant positioned on top to entice propagation and facilitate colonisation. *P.polycephalum* was then inoculated onto the reference electrode and left to forage in a dark enclosure. We programmed our system to take readings from each electrode at two-second intervals, and to capture images every two-minutes. Presented in Fig 9 are graphs produced by our system representing each measurement electrode's activity.

This collation process took five days to complete and generated an excessive 330,000 electrical readings for each measurement electrode. Harnessing this quantity of data would have resulted in an extremely long and undynamic output. To circumvent this, we applied both of our software's compression algorithms. First, sets of ten readings were averaged. Then, we only retained readings if they presented a change over a defined threshold. This process reduced the quantity

Fig. 7. Images of the step sequencer growth environment.

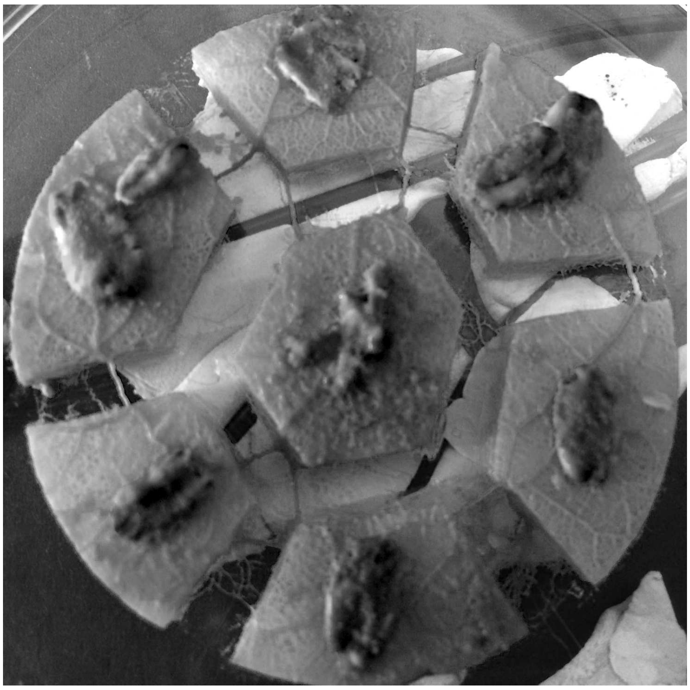


Fig. 8. *P.polycephalum* fully grown within the step sequencer growth environment.

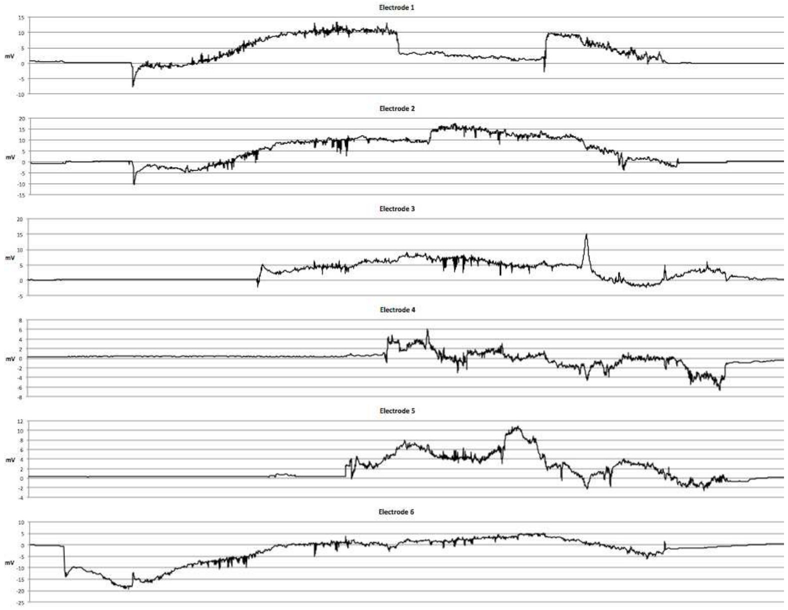
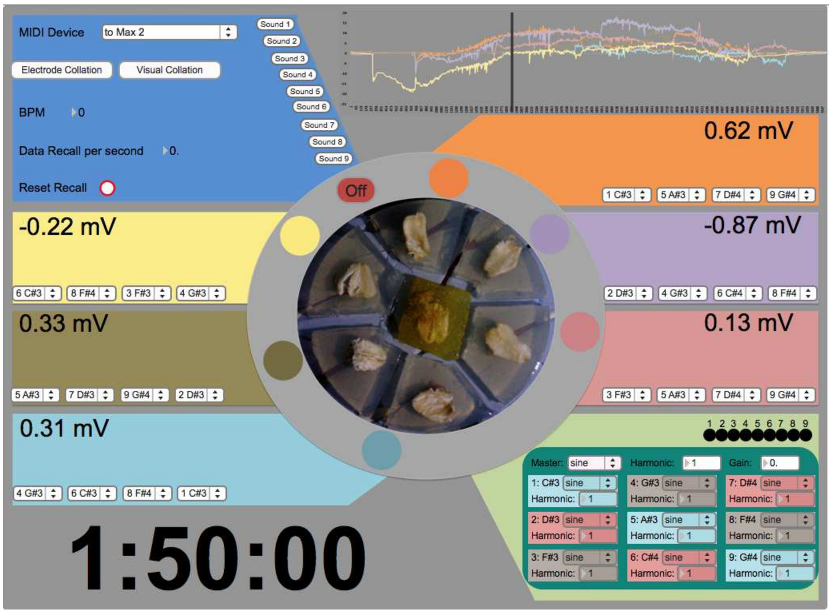


Fig. 9. A summary of electrical activity recorded on each of the six steps over the course of one week.

of entries to circa 5500, while maintaining behavioural patterns and the voltage gradients between them.

The step sequencer itself was programmed in Max with a small amount of data handling operations being outsourced to Java. This application's architecture mimics the growth environment used to record behaviour in and operates as follows. Six steps, arranged in a circular fashion, are paired with their respective electrode collation. These collations are recalled at a user-defined speed and are altered to become absolute values. A global metronome then ticks through each step at a user-defined BPM in a 360 loop taking a reading from each data stream. Steps only become active within the sequence once populated by *P.polycephalum*. Until this time, no reading is taken. Step activation is achieved by the application initially applying a high level smoothing expression to each data stream whose output activates the step when it reaches a threshold above zero. Readings taken by the sequencer are used to trigger a set of nine MIDI notes that are programmed in by the user. All steps are allocated a set of four notes from the nine available, which are then each assigned to a voltage trigger range. When a note is triggered, its velocity is produced through scaling the step's current electrical potential value to the MIDI data range. In order to determine the duration of a note, the current average potential of all other steps (active and non-active) is calculated, and then compared to the triggered step's voltage to produce a potential difference value. The higher this value, the more significant the note duration will be within the sequence, with a maximum duration being four beats. The sequencer is limited to only allow six notes to sound at a time; if a note is triggered but is unavailable due to being made active by another step, a note with the closest value in the step's priority list will sound. All of the sequencer's parameters can be altered in real-time.

As with any musical device, the user interface is an integral part of its function. Within this sequencer, the interface is built around the time-lapse imagery, inducing a connection between the user, the recorded behaviour, and the resulting sonic output. Here the time-lapse imagery is played back in perfect synchrony to the electrical collations and is positioned in the centre of the user interface with each step's parameter controls arranged next to their corresponding electrode (Fig 10).



MIDI Device: to Max 2

BPM: 0

Data Recall per second: 0.

Master: sine Harmonic: 1 Gain: 0.

1: C#3 sine Harmonic: 1
4: G#3 sine Harmonic: 1
7: D#4 sine Harmonic: 1
2: D#3 sine Harmonic: 1
5: A#3 sine Harmonic: 1
8: F#4 sine Harmonic: 1
3: F#3 sine Harmonic: 1
6: C#4 sine Harmonic: 1
9: G#4 sine Harmonic: 1

Fig. 10. The *P.Polycephalum* step sequencer user interface.

This sequencer outputs a progressive arrangement of notes, which correspond morphologically to the recorded foraging behaviour. As a sonification of behaviour, the produced output does convey auditory representation of voltage levels quite accurately. This is because each note's velocity is produced directly by the respective step's voltage - a parameter that directly relates to energy. Moreover, by producing each note's duration with a potential difference value, it is possible to compare activity on each step through listening. From a musical perspective, having note velocity controlled this way is slightly undynamic due to voltage levels also controlling which notes are triggered: each note is played with a similar velocity every time. Going forward, this parameter may need to be controlled by another behavioural aspect that provides an additional dimension to the sonification.

Although currently this device's implementation is simplistic, a range of different results can be achieved through utilising the available parameters. However, the quantity of original material produced through the extended use of this device maybe limited due to the application using only a single set of recorded behaviour. Future versions of the sequencer could be developed to embody a model of *P.polycephalum*, allowing for different behaviour to be produced and harnessed in real-time.

6 Final Remarks

This paper presented the development of a system for recording the behaviour of *P.polycephalum.* The purpose of such a system is to enable a set of sonification experiments to be completed, marking the start of our investigation into how the area of unconventional computing may be used in computer music.

At this time it is difficult to evaluate the success of our system due to its limited employment. The testing scenario did however give a preliminary indication that it was operationally sound. Also, our early experimentation does suggest a degree of usability. This was demonstrated by the step sequencer, which was implemented using data collected by our system without any major compatibility issues occurring. Moreover, its output gave a basic auditory perception of both *P.polycephalum's* developing morphology and electrical activities, indicating that our system recorded behaviour with levels of accuracy. To speculate on issues and limitations that could arise; the electrical potential collation frameworks may cause some problems. This is in regards to the quantity of data accumulated from recording behaviour. Such large quantities of information may be overly excessive to employ, and any compression applied could damage the integrity of recorded behaviour. If any problems such as this occur, we will improve our system as necessary. Moreover, as advances are made in the field of unconventional computing with *P.polycephalum*, we will review our system to ensure we are best positioned to take advantage of these.

In regards to our continued work in this area, we are extensively conducting experiments that harness the behaviour of *P.polycephalum* in several different ways. As these experiments develop, success levels will indicate areas of computer music that may benefit from the unconventional computing paradigm. We are also currently researching methods of controlling *P.polycephalum's* behaviour in order to develop degrees of predictable control during its use. This is an active area of research both in the field of biology and unconventional computing. Advances made in this area will improve *P.polycephalum's* compatibility with employment in computer music by allowing for behaviour to be manipulated with musical/creative intent.

In summary, our research is still very much in its infancy, but so is unconventional computing with *P.polycephalum.* To begin understanding how this branch of computer science may be used in music, we need to immerse its application across the field. This process will widen our appreciation and lead to innovative advances as the computing paradigm lends itself to certain applications.

Undoubtedly, further research and development into unconventional models of computation will be innovatively fruitful for the field of computer music.

References

1. Adamatzky, A.: Physarum machines: computers from slime mould, vol. 74. World Scientific (2010)
2. Adamatzky, A.: Physarum machines for space missions. Acta Futura 6, 53–67 (2013)
3. Adamatzky, A.: Towards slime mould colour sensor: Recognition of colours by Physarum polycephalum. Organic Electronics 14(12), 3355–3361 (2013)
4. Adamatzky, A., Costello, B.D.L., Asai, T.: Reaction-diffusion computers. Access Online via Elsevier (2005)
5. Adamatzky, A., Jones, J.: Road planning with slime mould: if Physarum built motorways it would route M6/M74 through Newcastle. International Journal of Bifurcation and Chaos 20(10), 3065–3084 (2010)
6. Adamatzky, A., Jones, J.: On electrical correlates of Physarum polycephalum spatial activity: Can we see Physarum Machine in the dark? Biophysical Reviews and Letters 6(01n02), 29–57 (2011)
7. Aspray, W.: John von Neumann and the origins of modern computing, vol. 191. Mit Press Cambridge, MA (1990)
8. Copeland, B.J.: The modern history of computing (2008)
9. Doornbusch, P.: Computer Sound Synthesis in 1951: The Music of CSIRAC. Computer Music Journal 28(1), 10–25 (2004)
10. Hiller, L.A., Isaacson, L.M.: Experimental Music; Composition with an Electronic Computer by Lejaren A. Hiller, Jr. and Leonard M. Isaacson. New York, McGraw-Hill (1959)
11. Ladd, T.D., Jelezko, F., Laflamme, R., Nakamura, Y., Monroe, C., OBrien, J.L.: Quantum computers. Nature 464(7285), 45–53 (2010)
12. Meyer, R., Stockem, W.: Studies on microplasmodia of physarum polycephalum V: Electrical activity of different types of microplasmodia and macroplasmodia. Cell Biology International Reports 3(4), 321–330 (1979)
13. Miranda, E.R., Bull, L., Gueguen, F., Uroukov, I.S.: Computer Music Meets Unconventional Computing: Towards Sound Synthesis with In Vitro Neuronal Networks. Computer Music Journal 33(1), 9–18 (2009)
14. Miranda, E.R.: Granular Synthesis of Sounds by Means of a Cellular Automaton. Leonardo 28(4), 297–300 (1995)
15. Miranda, E.R.: Composing Music with Computers, vol. 1. Taylor & Francis US (2001)
16. Miranda, E.R., Wanderley, M.M.: New digital musical instruments: control and interaction beyond the keyboard, vol. 21. AR Editions, Inc. (2006)
17. Nakagaki, T., Yamada, H., Tóth, A.: Intelligence: Maze-solving by an amoeboid organism. Nature 407(6803), 470 (2000)
18. Nakagaki, T., Yamada, H., Ueda, T.: Modulation of cellular rhythm and photoavoidance by oscillatory irradiation in the Physarum plasmodium. Biophysical Chemistry 82(1), 23–28 (1999)

19. Stepney, S.: Programming Unconventional Computers: Dynamics, Development, Self-Reference. Entropy 14(10), 1939–1952 (2012)
20. Tsuda, S., Zauner, K.P., Gunji, Y.P.: Robot control with biological cells. Biosystems 87(2), 215–223 (2007)
21. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London mathematical society 42(2), 230–265 (1936)