

Zombitron: towards a toolbox for repurposing obsolete smartphones into new interactive systems

Clara Rigaud
allo@clararigaud.com

Abstract

This article explores the possibilities of reusing obsolete smartphones and tablets to build new interactive systems. Taking the case of a musical instrument, I present my research into the design of a controller made from various of these obsolete smartphones. From the diagnostic stage to the creation of a new autonomous electronic object, I document the process, the barriers and the levers encountered. Based on these explorations and discussions with two professional musicians, I provide several insights into the software and hardware aspects, with a view to continuing this work, towards the creation of an open-source toolkit enabling anyone to build new interactive systems with old devices. I discuss the implication of how a high-level web-based approach could allow designers to enter the black box and foster permacomputing using smartphones.

Keywords

Smartphones, obsolescence, e-waste, toolkit, zombitron, permacomputing, NIME, demoscene, music, controller

1 Introduction

As electronic devices become increasingly widespread across the globe, mountains of electronic waste are reaching new heights. A recent report of the Global E-waste Monitor [6] estimates that 62 billion kg of electronic waste has been generated worldwide in 2022. Of this 62 billion, only 22% is documented as being recycled in a controlled and environmentally-friendly way and 22% is stored as waste in open dumps. The remaining 56% is recycled in uncontrolled circuits, which may release materials such as mercury or plastic into the environment exposing people to unhealthy work and unsanitary conditions [37].

Smartphones and other communication devices account for 8% of this waste. They are often abandoned because the battery no longer works, the screen is broken, they do not have enough memory space, the software is obsolete or they are replaced with more recent technology [30, 35]. Most of the time, these devices are kept by their owners and are not recycled, repaired or reused [35]. This means that there are plenty of phones lying dormant in the drawers of millions of people [3, 46].

Of course, work is being done on how to reduce this waste generated by smartphones. In particular, the main approach is to extend their lifespan, which is not always easy, given that many manufacturers make their equipment obsolete, and even base their business models on this obsolescence. More recently, legislative constraints have been implemented, and initiatives such as the reparability index or durability index have appeared in France [1, 2, 4] to limit this obsolescence, but this only applies to the most recently built devices, leaving a whole range of these smartphones

out of use. Another way of extending the life of smartphones is to refurbish them and buy them second-hand when the owner wants to get rid of them, but this only concerns the most recent models as well, leaving obsolete models aside.

The case I am interested in is the one where technologies are still functional but have been rendered obsolete anyway because they cannot keep up with current usage. For example, because they only have 3G, they're no longer powerful enough to support the latest software updates, or their storage memory is saturated [23, 30]. Although there are ways of prolonging the use of one's smartphone, prolonging the life of an obsolete device requires compromises in terms of use or advanced technical knowledge.

In this article, I propose a new approach consisting of combining several obsolete smartphones with each other to enable the creation of a new interactive system. I explore the possibilities that old devices can offer with an approach as high-level as possible, that does not involve reinstalling a custom OS, and that does not require a high degree of coding knowledge. After a brief overview of the existing works, I present the *Zombitron* project, which consists of setting up a toolbox of design and open-source software, and detail my explorations in the construction of this toolbox. Then, on the basis of these initial explorations, I document my research into the design of a new interactive system from the very first stage of getting it up and running. Using 14 discarded smartphones, I detail step-by-step the operations and problems encountered, and how these can be work-around, depending on the version of the operating system (OS), their performance and their general state of repair. Then, I present the discussions I had with two professional musicians on how this tool could be integrated into their work and practice, giving promising leads as to the appropriation of this approach. Finally, I explore the possible directions this work could take, in particular in the context of musical interfaces, to enable rapid prototyping, or as a pedagogical tool.

2 Background and related work

Smartphones and other communication devices account for 8% of the total of E-waste generated worldwide [6]. Obsolescence is the main reason why connected devices end-up being discarded, often because of limitations in terms of storage memory, computing capacity or communication protocols, which no longer allow the device to function in a wider technical ecosystem [8].

In addition to obsolescence, smartphones and tablets, compared to other communication devices, are also affected by a phenomenon of frequent renewal linked to the evolution of the perception of the value of their device by their owner [23]. The average length of use of smartphones by their first owner is estimated at 3.5 years in Western Europe [23]. Smartphone obsolescence is particularly linked to rapidly evolving technology and therefore to software that becomes more demanding in terms of resources and is only

maintained a few years after its release. For example, in the case of Apple products, the latest OS version supported at the time of writing is iOS 18¹, and the last device to be able to install version 17 is the iPhone XS² released in 2018. This means that all Apple smartphones released before 2018 can no longer get updates (including security updates). Android devices suffer from the same problem, with the difference that it is possible to install more recent versions of Android on older devices, which are more resources demanding and may not be suited to the configuration of an older phone. Generally speaking, the applications that are loaded onto smartphones and tablets are becoming more and more memory- and compute-intensive, slowing down devices and limiting their overall usability [30].

The main problem with smartphones renewal, is knowing what to do with them once they are no longer in use. The first approach is to reintegrate them into recovery circuits, either by recycling them, which recovers the rare metals that make up these devices (and therefore reduces the devastating extraction of these components), or by extending their life through refurbishing services or reuse strategies [35]. Refurbishing smartphones is a quite popular strategy since it has advantages for both manufacturers and consumers, but it is only reserved for the most recent products, and does not guarantee the lifespan or reliability of the refurbished device [16]. Furthermore, the recycling and reuse circuits are not always beneficial from an environmental and health point of view due to the infrastructures and standards in different parts of the world [6] and the rebound effects linked to energy expenditure and greenhouse gas emissions [24, 47]. In addition, many of these devices are not integrated into the recycling or refurbishing circuits at all. Wilson et al.[46] found in their 2017 study that over 50% of participants keep at least one mobile phone at home that they have replaced, usually “as a spare”. In France, ADEME [3] estimates that 1 million smartphones are lying dormant in French people’s drawers.

Studies have explored how to extend the life of obsolete devices, through alternatives and strategies for continuing to use them as their main smartphones. For example, Mosesso et al.[30] have studied the different reasons and strategies used by people who continue to use their obsolete smartphones. Goodwin et al.[13] explored the availability of applications on obsolete Apple devices. These two studies conclude that it is possible to find ways of continuing to use ageing smartphones, but that these strategies often require compromises or extra effort that not all users are prepared to commit to. Some of these techniques may as well involve extra skills, for example to install new alternative OSes³ on Android smartphones.

Another approach to reuse is to repurpose obsolete smartphones [7, 15, 47] so that they can be specialised to a precise task, better suited to their performance. For example Switzer et al.[43] explored techniques to repurpose smartphones into specialised servers for cloud computing. Klugman et al.[20] and Norbistrath et al.[31] explored how to turn them into gateways for IOT. Smartphones and tablets can also be repurposed to be dedicated to a particular application downloaded into it, for example in educational contexts [21].

The main obstacles in this repurposing lie in the heterogeneity of devices [21] and the relatively expert operations involved, if this repurposing goes beyond simply downloading a dedicated application. In addition, Apple smartphones are poorly represented in smartphone repurposing projects as they do not allow the installation of third party applications.

Repurposing techniques are popular in the Do It Yourself and hacking communities [22, 45], and are attracting widespread interest from researchers in sustainable human computer interaction [18]. In particular, the burgeoning permacomputing [25] movement is interested in what can be done with what exists in terms of technological hardware, and in particular in creation. This movement brings together not so recent approaches of creation but which, now, in the current context, are tinged with a particular urgency. Reuse in design has been a particular focus of Nicolas Nova [33], who presents the reuse of old technologies as a low-tech approach to design. Some new media artists are incorporating the reuse of dysfunctional and obsolete objects into their artistic practice, in line with the Zombie-media concept [17]. This is the case, for example, of Recyclism⁴, who have been exploring the revival of obsolete technologies for many years now. Others use electronic waste in their work to send out a message about technological consumption [5]. The permacomputing approach is also closely linked to that of demoscene, which since the 80s has been exploring the most restrictive ways of creating animations or music on the few remaining bits available on the first game consoles, and still today continues to gather around “release parties” where everyone comes to take part in demo⁵ competitions in their own category (weight/platform).

While these DIY communities are reputed for their openness and habit of sharing knowledge and technique [39, 45], the fact remains that these practices are reserved for relatively expert users. Electronic objects have become smaller and complex [11], and the smartphone has become a little box filled with different sensors that are difficult to reuse and tamper with [32]. A few soldering points are no longer enough to hack and bend must devices, leaving the reuse of obsolescent smartphones relatively absent from these projects.

In response to the increasing complexity of electronic devices, which makes certain sustainable practices difficult for the general public to access, Roedl et al.[42] encourages HCI researchers to develop tools to help with the appropriation of these practices and techniques. Similarly, Lu et al.[22] argues for a field of HCI research focusing on the design of tools for recycling, recovery and reuse.

In a nutshell, there are initiatives and contexts in which the reuse of obsolete smartphones can be explored. But the complexity of these devices and their heterogeneity require advanced technical knowledge, which can act as a barrier to their reuse. It is therefore crucial to design tools to encourage and facilitate the appropriation of techniques for repurposing these obsolete devices, and fully integrate obsolete smartphones in the permacomputing garden.

3 Zombitron project and principle

The aim of this article is to present the initial work on the design of new interactive systems using a high-level approach through the

¹<https://endoflife.date/ios>

²https://en.wikipedia.org/wiki/IOS_version_history#Hardware_support

³https://en.wikipedia.org/wiki/List_of_custom_Android_distributions

⁴<https://www.recyclism.com/>

⁵<https://www.pouet.net/prodlist.php>

Zombitron project. The aim of *Zombitron* is to enable new systems to be designed from obsolete smartphones and tablets, taking into account their heterogeneity. Given that these devices offer many possibilities in terms of connectivity and interactivity, the objective is to explore how we can continue to benefit from them. *Zombitron* aims to make the design of new systems based on these devices accessible by facilitating access to the functionalities of the black box that are smartphones and tablets, and to become an open-source and collaborative tool enabling artists and everyone to address these issues of obsolescence in the case of the smartphone.

3.1 Principle

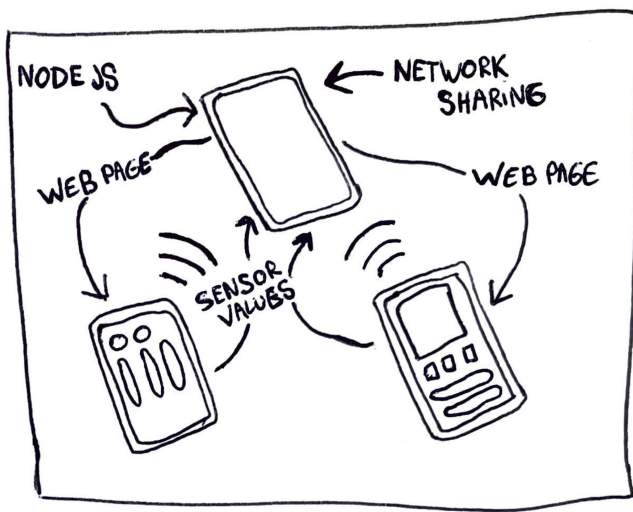


Figure 1: The basic idea behind *Zombitron* is to connect several smartphones together to share their information.

The *Zombitron* principle is based on the fact that whatever the release date of a smartphone and its OS, it has a WiFi card and a browser. From there, it is possible to connect these smartphones and tablets to a server that displays web interfaces and exchanges information via the WiFi connection.

As shown (Figure 1), the system is composed of different units that can be any connected device with a browser:

- (1) A unit that acts as a hotspot so that others can connect to it locally.
- (2) A unit that runs a terminal enabling a *Node.js* instance to be run and an http server to be generated on this network, also enabling the websocket protocol.
- (3) One or more units that will connect to the server as web clients.

For example, depending on its configuration, a unit can act as a hotspot (1), run the server (2) and display a web interface (3). It is also possible to use a computer to run the server and an external router to create a local network.

The way in which the system is implemented will essentially depend on the capabilities of each of the units:

3.1.1 Creating a local network. There are several ways of creating a local network. The simplest way is to use a dedicated router, for example an internet box. But in the case of an interface designed to be autonomous and operate without any devices other than the smartphones and tablets it is made up of, a shared connection needs to be set up. To do this, it is possible to activate tethering mode on certain devices. This option is widely available, but some OSes do not allow this option to be activated if there is no access to mobile data. The unit bundle must therefore include at least one device enabling tethering without mobile data.

3.1.2 Running a server with Node.js. To run a *Node.js* server, one of the units needs to be able to run a terminal. To do this, *Zombitron* uses the *Termux* application, which emulates a Linux terminal on Android devices. This enables packages to be downloaded using “apt” and *Node.js* to be run. With *Node.js*, an http server is set up to deliver webpages to other units connected to the network, and a websockets server is used to send data between units in real time.

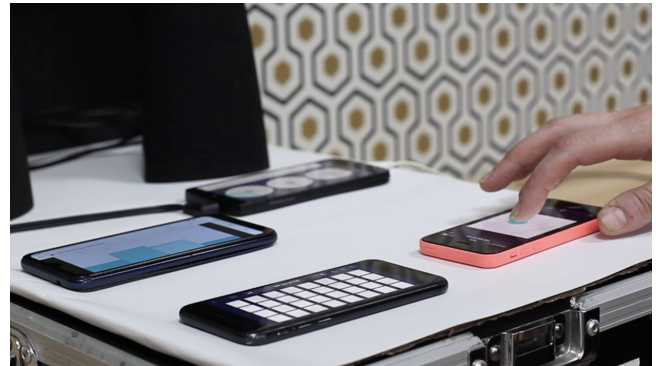


Figure 2: This images shows how one can create a new interface by loading webpages on different smartphones.

3.1.3 Displaying web interfaces. Once this environment has been set up, it is possible to set up any web application that is capable of being executed on the smartphone browser. For example, figure 2 shows the control interfaces loaded on different smartphones composing music. Each smartphone displays a webpage and can send messages related to touch events to the other units.

3.2 First explorations

Based on this principle, two musical instrument prototypes have been designed: *Zombitronica* and *Zombichord*. These two prototypes are built using the *Tone.js* library dedicated to sound synthesis through web audio [41]. From a hardware point of view, both are built from reused materials, and operate in standalone mode.

3.2.1 Zombitronica. *Zombitronica* is the first prototype based on 4 smartphones and a loudspeaker, all second-hand. The interface has been designed to explore different types of touch controllers for generating music (Figure 3). It is made up of a sequencer for selecting the notes played by the 4 instruments over 8 beats, 4 sliders for controlling the volume of each of the instruments, a two-dimensional slider for controlling a 5th instrument (its pitch



Figure 3: Zombitronica is the first prototype exploring the principle of Zombitron, made with 4 discarded smartphones and a loud-speaker.

and volume) and 3 potentiometers controlling the speed of the metronome, distortion and reverb. From a hardware point of view, all the phones are powered by a usb hub, and the phone generating the music is connected to the speaker via a 3,5mm jack. A laser-cut sheet of 3mm PMMA delimits the screen areas corresponding to the control interfaces, and holds the 4 smartphones and the loudspeaker in place. Each component rests on a plate suspended from the top by a system of screws, nuts and spacers.

3.2.2 Zombichord. *Zombichord* is a second prototype inspired by the Omnichord⁶ and built using an Android 5 tablet and an IOS smartphone. On this prototype, the *Node.js* server is run on the tablet, which also displays an interface for playing chords, and an iPhone 6 displaying an interface for playing notes on the scale corresponding to that chord [figure 4]. For the hardware, a thick piece of wood was hollowed out with a wood router to accommodate the two devices. Holes were drilled to allow the lightning-type cable connected to the iPhone to pass through on one side and a mini usb cable to supply power to the tablet on the other. The lightning cable is used to output both the sound signal and to supply power to the phone, so a few soldering points were used to make a small assembly that simply reveals a female jack (for the sound output) and a female coaxial socket for supplying 5V to the two devices.

3.3 Outcome

These two prototypes enabled me to test a web approach to creating two musical interfaces on discarded smartphones and tablets. Although only the touch screen was used, they already enabled me to identify certain opportunities and limitations linked to the age of these devices, particularly in terms of performance and functionality. The choice of implementing the software in webpage format on older smartphones, which have less computing power than today's products, may mean that we have to use lightweight applications. Devices that only load a touch interface and send their data via



Figure 4: Zombichord is the second prototype exploring the principle of Zombitron. It was designed by Romain Segaud and Marion Jolas, inspired by the Omnichord.

websockets did not pose any problem, but the use of a library that requires a little more CPU processing, such as *Tone.js* doing sound synthesis, should be considered for rather more powerful smartphones. In order to eliminate the latency of the two prototypes, the use of *Tone.js* was restricted to the activation of pre-recorded samples. This opens up opportunities to explore how the combination of older and more efficient equipment can overcome these limitations. The second prototype was designed using a tablet running Android 5.0. For this tablet, the *Termux* application was no longer supported, which means that the package database is no longer updated, limiting the version of *Node.js* to version 12. A few readjustments to the code enabled me to build the application with *Node.js* 12 without any problems.

4 Studying the possibilities offered by obsolete smartphones

On the basis of the two explorations made through the *Zombitronica* and *Zombichord* prototypes, I have sought to confront this approach with a more general situation, to consider the limits that a variety of these old devices may have, and the ways in which they can be overcome. With a view to creating a common tool that would enable everyone to design with obsolete devices, I looked at a wider range of these devices to document, step by step, the specifics in terms of installation and configuration that I may encounter. I collected a set of 14 smartphones that had been given to me or that I had bought cheaply second-hand. For each of the phones, I went through the different stages to set up the basis of the *Zombitron* principle. This time, in addition to the touch sensor, I integrated the Inertial Measurement Units (IMU) sensors: gyroscope and accelerometer, common to most smartphones since 2012.

During this exploration, I set up a common code base on a git repository⁷, allowing me to run tests on the different devices, and which I improved as I went along so that it took into account the

⁶<https://en.wikipedia.org/wiki/Omnichord>

⁷The code base used in this article is preserved here: <https://zombitron.clararigaud.com>

particularities of each one. This code can then be used as the basis for all interfaces created using the *Zombitron* principle, and can be expanded as we implement them. Additional resources on specific commands I have used can be found in Appendix A.

In this section, I break down the steps from the first switch-on to loading the webpage and sending the sensor data, documenting the obstacles I encountered depending on the device.



Figure 5: Overview of the 14 discarded smartphones batch.

4.1 Turning on and unlocking

	Model	Year	Server	Hotspot	Client
(1)	iPhone 5	2012			■
(2)	SM Galaxy S III	2012		■	■
(3)	SM Galaxy S III	2012			
(4)	Moche smart A8	2012		■	■
(5)	iPhone 7	2016			■
(6)	SM Galaxy J5	2017	■		■
(7)	Huawei P8 lite	2017	■	■	■
(8)	Moto E4	2017			
(9)	Gigaset GS80	2018	■	■	■
(10)	Archos oxygen 57	2019	■	■	■
(11)	Huawei Y6	2019			
(12)	Fairphone 3	2019			
(13)	Alcatel 1B	2020	■	■	■
(14)	Oppo find x2 lite	2020			

Table 1: List of the 14 smartphones selected with their year of release and their ability to be used as a server unit, a hotspot unit and a client unit. Devices that could not be turned on or unlocked are marked in pink.

The first step for each smartphone is to turn it on and unlock it. Some of the phones in my batch had not been switched on for a long time, or had been considered “bricked” by their users. So for each phone, I began by charging the battery, then tried to switch it on. Some did not make it through this stage, and so I eliminated them from my batch as they require a more advanced level of repair. Table 1 summarises which of the 14 smartphones could be turned on and unlocked, 5 smartphones were eliminated at this stage. For example, this was the case with one of the two Samsung Galaxy

S III (3), the Huawei Y6 (11), Moto E4 (8) and the Oppo Find X2 (14). (3), (8) and (11) had no reaction when the power button was pressed, requiring a diagnosis beyond my scope. On the other hand, (14) vibrated when switched on without displaying anything, suggesting a dead screen. To check whether the devices are detected as serial usb ports (and therefore check whether they are alive), I connected the smartphones to my computer to verify whether they were detected as a USB serial interface. The Oppo (14) was indeed detected as an interface. An alternative rescue would therefore be to try connecting it to an external display using the “HDMI Alt Mode”⁸ protocol, but this was beyond my scope. Interestingly, the Fairphone (12) did not pass this stage either, because it was locked. After switching it on, I had to enter a password that I did not have, and I did not know the person who had given it to me. I naively tried to carry out a factory reset on the phone, but I came up against the Factory Reset Protection protocol, which requires signing in to the last Google account associated with the device. There are strategies, but they are time consuming, require scrolling many forums and “adventurous” operations, as I read on one of them. Some of these strategies involve exploiting security breaches that are patched over time, which means that the newer, or recently updated the phone, the less easy it is to bypass the FRP. So it is important to make sure that devices are reset to zero before disposing of them.

4.2 Setting up a server

The second step is to examine which smartphones are capable of acting as a server, which involves running *Node.js* in my case. To install and run *Node.js*, *Zombitron* uses the *Termux*⁹ application, which is only available on Android. As a result, the two iPhones (1) and (5) in my selection will not be able to be server units.

There are several ways to download *Termux*, the easiest of which is to go to the Google Play Store¹⁰ and download the application, but this is only available for devices with Android 11 (released in 2020) or later. The other disadvantage is that this involves associating all devices with a Google account. For these reasons, I chose to install the application without going through the Google Play Store, but rather directly by downloading the .apk file. This process requires a few additional steps: installing *Android Debug Bridge* (adb)¹¹, and enabling USB debugging mode on the device to be able to load applications on the phone. These steps are widely documented on the web. Once this is done, it is possible to easily access the smartphone’s information and install apk files on it. The next step is to download the apk file, for example from *Termux*’s github¹², selecting the file corresponding to the Android version and the device’s CPU ABI¹³. Table 2 gives a summary of the different versions and CPU ABI’s of the Android devices in my batch.

Once *Termux* is installed and running, it is possible to install *Node.js*. (The smartphone must be connected to the internet first.) For devices running Android <7, the version of *Termux* no longer

⁸<https://www.hdmi.org/spec/typec>

⁹<https://termux.dev/en/>

¹⁰<https://play.google.com/store/apps/details?id=com.termux>

¹¹<https://developer.android.com/tools/adb>

¹²<https://github.com/termux/termux-app>

¹³Android Binary Interface: <https://developer.android.com/ndk/guides/abis>

	Android version	Processor's ABI	Termux	Node.js version
(2)	6.0.1	armeabi-v7a	■	12.13.0
(4)	2.3.6	armeabi		
(6)	9	armeabi-v7a	■	22.14.0
(7)	8.0.0	arm64-v8a	■	22.14.0
(9)	8.1.0	armeabi-v7a	■	22.14.0
(10)	9	arm64-v8a	■	22.14.0
(13)	10	armeabi-v7a	■	22.14.0

Table 2: Summary of the Android versions, the ABI of each of the android Smartphones in the batch, the availability of Termux and, if applicable, the version of Node.js available.

updates the package database, which sometimes forces the use of discarded version packages¹⁴. The installation of *Termux* and *Node.js* went well on the device (2) running Android 6. The version of *Node.js* available in Long Term Support (LTS) is 12.13.0. However, an error occurred when using the `npm` command, which allows packages to be installed in *Node.js*. Unfortunately I did not have any other devices running Android 5 or 6 to compare, but as the tablet used in the *Zombichord* prototype runs Android 5, I know it is possible to use these OS versions. *Termux* is available from Android 5, which eliminates my device (4) running Android 2.

Once *Node.js* was installed, I used `git` to download my test environment. At this stage, the folder only contains a *Node.js* project and the necessary dependencies for *Zombitron*: `express` to create `http` servers, and `websocket` to use the real-time socket protocol for sending messages. The aim is to check that all versions of *Node.js* can generate servers with these two components, and if there are any particularities to take into account depending on the version of *Node.js*, to ensure that these are added to my code. As a result, and after a few syntax corrections linked to the different versions, both version 12.13.0 and the current LTS version (22.15.0) of *Node.js* can run the server.

4.3 Setting up a local network

For the interface built with *Zombitron* to be standalone, at least one of the devices must be able to act as a hotspot to generate a local network among all the devices. While most devices allow this connection sharing option, some OSes require mobile data to activate this network sharing. This is the case for all IOS devices, and some Android-based interfaces like in my case here with device (6) which has a One UI 1.1 interface. Table 1 summarises the devices in my sample that can be used as hotspots.

4.4 Setting up web clients

To test the possibilities of using each of the smartphones as a client, I developed, as in the case of the server, a code example which I put to the test on each of the devices and their browsers, and I improved it according to the problems encountered. I managed to get a touch interface working and to obtain IMU data from all the

phones and send it to the server via `websocket`. I detail below the obstacles encountered and how I overcame them.

Javascript ES6. To communicate with the server, clients use the javascript language. In 2015, a major version of javascript (ES6) was released, making some older browsers incompatible. This is the case, for example, with the browsers of (4). All the interfaces developed on *Zombitron* are compatible with the older version of JavaScript (ES5).

Motion sensors. Most smartphones have two inertia sensors: the accelerometer and the gyroscope. In my batch, all the devices have an accelerometer, and all except (7) (9) (10) (13) a gyroscope. These two sensors respectively provide motion and orientation data via the `DeviceOrientationEvent` and `DeviceMotionEvent` interfaces.

I was able to obtain sensor data from all the devices, taking into account the following small details in my code.

- Safari on IOS 14.5 introduced a new security feature for access to inertial sensors: `requestDevicePermission`.
- Access to sensor data requires a secure server (`https`) on most recent browsers, which I have enabled as an option in my code.

Secured server (https). A secure server is set up by creating a key and a certificate. Each certificate is linked to the server's IP address. As secure servers are necessary for the use of inertial sensor data, I have created a script allowing certificates to be generated directly from a command line.

Very old versions of Android Browser before 4.4.4 are incompatible with the `https` protocol. To avoid this problem on (4), I installed a version of Firefox (Firefox 31) that supports `https`. The implementation of secure servers also has the effect of displaying alert pages when the interface is loaded, but these can be bypassed, often by clicking on a link such as "I trust this website".

Secured Websockets (wss). When the server is set for `https`, the `websocket` protocol must also be secured. This is taken into account in my code, but in the case of IOS devices < 13, it is necessary to install the certificates to use the `wss` protocol. To do this, the certificate needs to be downloaded, installed, then activated in the IOS settings: Settings > General > About > Certificate Trust Settings

4.5 Outcome

By taking a heterogeneous batch of smartphones, varying in their OS, release date and browser, I was able to explore in greater breadth the possibilities offered by a web approach such as *Zombitron*. This methodical and iterative work has also enabled me to develop a common code base that can run on all these devices, and that supports most of their specific features. This provides a base from which to create new applications and which can be fed with other functionalities and APIs.

5 Integration into two musicians' practices

While I have investigated the potential of a web-based approach to creating controllers from obsolete smartphones, from a technical point of view, I am now trying to understand how this tool could be integrated into real practices, using the example of music interfaces.

¹⁴<https://github.com/termux/termux-app/wiki/Termux-on-android-5-or-6>

During my study, I had many exchanges with two friends, both professional musicians with a background in electronic music composition and performance using numerous controllers and synthesizers. We exchanged ideas about their practices and habits in using tools to produce and perform. Then we imagined how *Zombitron* could enable them to build their instrument or their musical controller, and more generally how they envision the directions this tool could take.

Despite their very different practices and styles, Jean Turner¹⁵ and s8jfou¹⁶ both have a strong focus on tools and machines, which they like to make their own by exploring the possibilities of customisation. Both have a particular affection for the restrictions that machines bring to their work, and at the same time the freedom that computers bring to sound design.

Jean Turner's musical performances are somewhere between dance and experimental music, often using samplers and sequencers that he interfaces with his computer via *Ableton*¹⁷ and MIDI¹⁸. On stage, his gestures and movements are at the heart of his performance. Jean Turner has built instruments and controllers using "Axoloti core" [9] which he describes as a sort of audio and MIDI Swiss army knife, very simplified for the user. It allows him to do precise things according to his needs and to have certain functions that do not exist in other controllers.

Questioning the tools and the visual composition on stage is particularly at the heart of the work of s8jfou. s8jfou makes electronic music essentially with his computer and builds his instruments with MAX/MSP¹⁹ with which he interacts on stage. He has also built a few synthesizers²⁰ himself using micro-controllers such as *Teensy* [10] and *Raspberry PI pico* [14].

5.1 *Zombitron* as a performing tool for Jean Turner

When I presented my work on *Zombitron* to Jean Turner, he immediately imagined how he could integrate motion sensors into his scenic performance and composition. Using smartphones to perform opens up new horizons for him. *"I did not want to play with a computer on stage because I did not like the aesthetics. With Zombitron, it is logical because it democratises computer music even more and there is a playability that does not exist with other instruments."* Jean Turner also mentions the educational aspect that this can engender in his stage performance: *"For the audience, turning knobs does not mean much, whereas seeing the sound change with the tilt of an old smartphone is more intuitive."*

Jean Turner started building a whole range of articulated supports using LEGO® Technic to use smartphones as controllers connected to his Ableton software with Open Sound Control. As shown Figure 6, he created several supports to hold his two smartphones and then mounted these supports on different mechanisms, allowing them to rotate on two or three axes. Some of these mechanisms make use of motors to activate a continuous rotation of one



Figure 6: Jean Turner has created different mechanisms to design an interactive interface for playing with smartphones connected to his software via Open Sound Control.

smartphone and lights activated by the weight of the smartphone when put in certain supports. While he was making his LEGO®, he sent me a number of videos, in which he describes his enthusiasm as he explores the interactions, the effects he will be able to generate and the way he will compose his live show. *"I can put one on top of the other, and then my values are the same for both!"*. *"I just thought I can even use it as percussions."*

5.2 Zombee, a collaborative musical instrument for s8jfou

s8jfou immediately saw in *Zombitron* the possibility of creating a collaborative composition environment, and was stimulated by the idea of generating sound capable of being played by devices with very limited computing power.

In 2022, MAX released *RNBO*, a library for exporting MAX projects (patches) in portable code to compile them directly for given platforms, in particular WebAssembly. They have also released an npm package²¹ that allows this compiled code to be loaded directly via *Node.js*. From there, it was easy to imagine how to combine s8jfou's practice of MAX and *Zombitron*.

s8jfou has designed Zombee, which he began to implement in MAX. Zombee is an instrument consisting of 4 playing interfaces and a master interface, enabling several people to compose and play together. The 4 tactile interfaces (Figure 7 (A) and (B)) differ simply by the sample bank, and can be percussive, bass, synth, noise, etc. To this sound "search" interface, he added a sequencer to be able to compose. Added to these 4 instruments, he created an interface dedicated to the controlling the different parameters of the 4 instances (Figure 7 (C)), the volume, the BPM, the different effects (Delay, Reverb,...). *"A bit like an orchestra conductor"*. s8jfou envisaged Zombee by thinking about the limits in terms of memory and computing capacity that the device generating the sound might have. To achieve this, he uses the principle of granular synthesis

¹⁵<https://abstraction-mathematique.bandcamp.com/>

¹⁶<https://s8jfou.bandcamp.com/>

¹⁷<https://www.ableton.com/>

¹⁸<https://en.wikipedia.org/wiki/MIDI>

¹⁹<https://cycling74.com/products/max>

²⁰<https://www.s8jfou.com/synth.html>

²¹<https://www.npmjs.com/package/@rnbo>

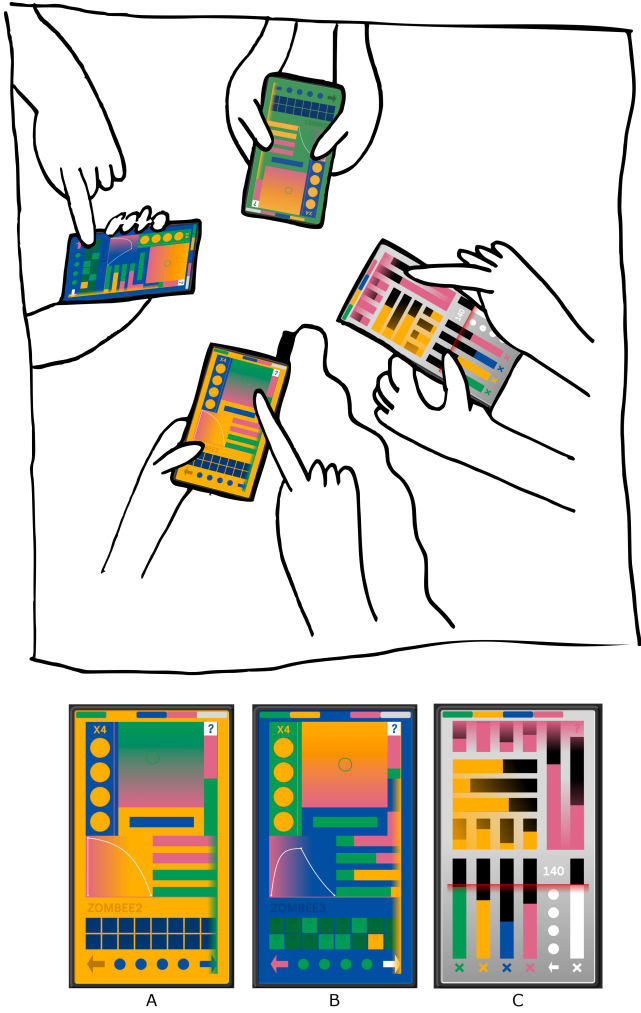


Figure 7: s8jfou has begun designing *Zombie*, a collaborative instrument optimised to run on devices with low processing capacity. A and B show two of the four composition interfaces and C shows the conductor interface.

[40], involving the use of samples of very short duration so that relatively light files can be loaded. He has designed his instrument so as to connect all the effects to a single output managed by the conductor, which also means that the effects (which consume a lot of CPU resources) are applied only once. He also identifies the advantage of versatility offered by this approach, imagining how it could be both stand-alone and used with a more powerful computer. *“Depending on the power of the device used, more or less can be done in sound synthesis. The tool is therefore very powerful if you want to use it, for example, with a recent computer but with old phones to control it remotely, but is also capable of doing more modest things when run from an old phone.”*

5.3 Outcome

These exchanges and the way in which s8jfou and Jean Turner envision the integration of obsolete smartphones into their work is very promising and opens up many avenues for reflection. The reuse of obsolete smartphones made accessible thanks to *Zombitron* has clearly stimulated my two friends, and largely motivates the continuation of this work, by developing specific tools to make their two prototype projects possible. I discuss the generative potential of these discussions and beginnings of work as well as the implications for tools and future work in the next section.

6 Discussion and future work

In this article, I explored how a web-based approach could enable the reuse of a heterogeneous set of obsolete smartphones and tablets varying in OS and release date. This work allowed me to explore how to create new interactive systems by combining devices of different natures as basic materials. This approach can be seen as an “augmentation” as described by Remy et al.[36], as this combination is an augmentation of obsolete smartphones with other obsolete smartphones.

During my experiments on 14 obsolete smartphones, I initiated the development of a code base that can be used as the starting point for various applications. In particular, I explored how this approach could be used in the work of musicians through discussions with s8jfou and Jean Turner. This study and exchanges open up some promising perspectives on ways of pursuing the work toward a toolkit and combining it with the various software tools used by musicians, other creative practitioners and beyond.

The whole idea behind this work is to define the position of the cursor between the design opportunities and the device’s capabilities. In terms of design opportunities, this work is primarily focused on the possibilities for interaction offered by the combination of several obsolete smartphones or tablets. In the code base I have developed for the purposes of this study, I have integrated tactile, motion and orientation data by exploring the different constraints generated by the different ages and operating systems of smartphones. For now, my approach has been to start with the older ones and assess what they can do, assuming that the newer ones can also do it. Therefore, the first straightforward way of pursuing this work is to explore the other interaction possibilities offered by smartphones. For example, with the camera or microphone, which could easily be streamed using protocols such as WebRTC. In my sample of 14 phones, many are also equipped with a brightness sensor, a fingerprint reader, an NFC chip detector, and many others which are all interaction opportunities with which to design new systems. Access to the installation of recent browsers on older devices will also make it possible to benefit from the latest APIs such as SensorAPI²², USB²³ or even bluetooth²⁴. The possibilities for interaction can also be extended by the simple fact that, by definition, any object capable of connecting to a network can be interfaced with *Zombitron*. It would be interesting to connect this work with the one of Norbistrath et al. [31] on repurposing smartphones into

²²https://developer.mozilla.org/en-US/docs/Web/API/Sensor_APIs

²³https://developer.mozilla.org/en-US/docs/Web/API/WebUSB_API

²⁴https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API

gateways for IoT. More generally, exploring the limitations of a web approach based on *Node.js* is another avenue of research.

In this work, I covered the fact that the capability of devices depends on their age and OS, and I explored these aspects in my study. However, capabilities also depend on other factors, in particular their general condition. Most of the smartphones I have used have been in pretty good condition, apart from some broken screens and batteries at the end of their life. It would therefore be important to continue this work, both by continuing to explore interactive capabilities, but also by confronting devices with possibly other limitations. Further work could also look at ways in which end-users can identify and choose combinations of obsolete devices based on their performance. And extend it to other, more application-optimal techniques.

The other major part of the design opportunities is the different applications that this tool base will enable. Here, there are two different directions which are perfectly illustrated by the two distinct approaches of Jean Turner and s8jfou. In one case, Jean Turner envisages his use of *Zombitron* as a controller connected to *Ableton*, while s8jfou designs *Zombiee* as a standalone. On the one hand, *Zombitron* must be capable of interfacing with the software and, on the other, the application is executed by the smartphone. These two approaches do not involve the same things.

In the case of Jean Turner, this is very easy to envisage as it does not require much extra work from the smartphones. *Zombitron* is based on *websocket*, but it is also possible to communicate sensor data from other protocols such as Open Sound Control (OSC), MQTT, and others that are relatively easy to implement in the tool with *Node.js*. OSC is a relatively widespread protocol, not only for controlling music but also other VJing or real-time graphic composition software such as *Touch Designer* or *Resolume*. It would also be interesting to explore how a controller made from discarded smartphones would be considered by AV practitioners. It would also be simple to turn it into a stream deck.

In the case of s8jfou however, the underlying challenge is to explore the capabilities of smartphones and see how to optimise applications and compose them specifically for computing performance. I was confronted with latency problems in the case of the two prototypes *Zombitronica* and *Zombichord*, using the Web Audio library *Tone.js*. At that point, the only strategy explored was to remove all the effects that consume resources. A very limiting approach in terms of functionality and creative possibilities for musicians.

This opens up avenues of exploration for future work on web approaches to performance-constrained audio synthesis on obsolete smartphones [44] and interactivity [28]. From a web perspective in particular, one strategy to explore is the use of *WebAssembly*²⁵, a format that allows most hardware capabilities to be accessed from the web browser. It is on this principle that *RNBO*, the portable extension of *MAX*, is based. *MAX* is a popular tool among sound creators and it would therefore be a good entry point for exploring *WebAssembly* on obsolete smartphones. The idea of *Zombiee* represents an exciting direction to explore in terms of how a combination of more or less powerful devices can generate rich music and allow musicians a wide playing environment.

The advantage of collaborating with practitioners is that, as well as enabling a new approach to be compared with real practices, there can be an overlap of optimisation approaches. The latter can be thought out by the tool implementation, but also by the creator themselves, for example here, the idea of using granular synthesis of s8jfou. The work initiated with *Zombiee* is also very promising in terms of the collaborative aspect that s8jfou envisages for this prototype, and its approach is very much in line with the work of Golvet et al.[12] and Matuszewski et al.[27–29]. It would also be interesting to combine low-tech approaches to instrument making with *Zombitron*, such as Nunes et al.'s ingenious ideas featured in Sibilm [34]. Sustainability seems to have a place in NIME's thinking [26], and there are undoubtedly many bridges to be built with the work of this community, given their thinking and political positions on technologies [19]. The case of music and art in general is a good research environment for this work since the communities are inclined to explore new approaches for creation. The musical axis allows many aspects at the heart of the problems of this work to be explored, namely interactivity, design and performance.

That said, this work could also have implications in other contexts, especially because of the low cost of implementation made possible by the reuse. For example, it could be used for low-cost installation in the case of documentation tools in makerspaces and laboratories [38], it could also enable rapid prototyping of interfaces and be used as inexpensive teaching material. s8jfou and Jean Turner both mentioned to me that it would be great to teach computer music in the workshop they or their friends are giving.

My discussions with s8jfou and Jean Turner also suggest a generative power of the reuse of smartphones in creativity. Firstly, from the point of view of the aesthetics of using smartphones. Neither Jean Turner and s8jfou did particularly envisage the smartphones being concealed in an outfit, as had been explored in the case of *Zombitronica* and *Zombichord*. On the contrary, Jean Turner explained to me how he planned to make these phones an integral part of his live show. “*It’s great, I will be able to take them and put them next to each other, then put them back down again.*”. He also mentions the benefit of a smartphone as an interactive interface known by the audience, enabling them to understand what is happening on stage, as opposed to the action of turning knobs. This relates to Mansoux et al.[25]’s reflections on retro-computing aesthetics, which tend to persist in permacomputing approaches. Perhaps we could imagine an antero-computing aesthetic embodied by the smartphone or other smartwatches in the next few years. Another interesting generative aspect is that, although the approach is by no means a new technology, the idea of building a new system with obsolete smartphones seemed to stimulate my two friends. For example, gyroscopes have been around for a long time, as have smartphones connected to local networks, so *Zombitron* is not inventing anything that does not already exist. But in the case of Jean Turner and s8jfou, it has generated a lot of new ideas. Perhaps the anticipation of a simple, accessible tool opens up a whole range of new possibilities.

Although this paper is essentially the beginning of work towards a toolbox to facilitate the replacement of obsolete smartphones, and my discussions were limited to two of my musician friends, it already identifies the opportunities that a collaborative approach between researchers and designers can generate for the design

²⁵<https://webassembly.org/>

of tools. This is in line with my desire to build bridges between communities, and the importance of designing tools to encourage the appropriation of sustainable techniques [22, 42].

My aim is to push this study further by exploring the design of new prototypes to explore more interaction opportunities and software and hardware limitations in a research-creation process as a means of feeding this toolbox. The projects we have initiated with s8jfou and Jean Turner are an embodiment of this.

7 Conclusion

In this article, I presented *Zombitron*'s simple idea of making interfaces with obsolete smartphones using a high-level, web-based approach. Then, I've explored this approach in more depth, confronting it with a range of obsolete smartphones varying in OS and software version. This enabled me to start the work toward a promising toolkit to favour appropriation of the reuse of these discarded devices. Through exchanges with two musicians leading to two speculative prototypes of musical controllers, we were able to explore how this toolkit could possibly be relevant to their practice. This work gives rise to a number of directions for research into the development of re-use techniques to encourage appropriation by creative practitioners and beyond. Firstly, it provides inspiration to pursue work on this toolkit and to continue exploring the design opportunities offered by the combination of obsolete smartphones and other obsolete devices. Secondly, it motivates the collaborative approach with artists to determine how the use of smartphones could not only enable them to adopt a more sober and less costly approach to their use of tools, but also be integrated as a central element of their performance. More generally, this work represents a new step towards democratising the use of obsolete smartphones, despite their complexity and heterogeneity.

Acknowledgments

All the research, including the state of the art, the material and methods, the original thinking, the writing of this article and the development of the code base I used for testing, i.e. everything described outside section 3 of this article, was carried out from 28 February 2025, by me if not otherwise specified, on my own time and with my own funds. The work on the two prototypes *Zombitronica* and *Zombichord* presented in section 3 was carried out while I was an employee of noesya, which retains ownership of them.

I would like to thank Marion Jolas for her help in making the *Zombichord*, Villette Makerz for hosting me as a resident during the construction of *Zombitronica* and *Zombichord*, La Generale for hosting me as a resident during the writing of this article and Jean Turner and s8jfou for the time they devoted to me, their interest and their help, which makes the future of this research so promising.

References

- [1] 2020. Décret n° 2020-1757 du 29 décembre 2020 relatif à l'indice de réparabilité des équipements électriques et électroniques.
- [2] 2024. Décret n° 2024-316 du 5 avril 2024 relatif à l'indice de durabilité des équipements électriques et électroniques.
- [3] ADEME. 2023. *Longue vie à notre smartphone*. Technical Report 011707. ADEME. 7 pages.
- [4] ADEME, In Extenso Innovation Croissance, Benoit TINETTI, Marion JOVER, Chloé DEVAUZE, Mariane IGHILAHRIZ, Fraunhofer IZM, and Anton BERWALD. 2021. *Preparatory study for the introduction of a durability index*. Technical Report. ADEME. 180 pages. <https://bibrairie.ademe.fr/economie-circulaire-et-dechets/4853-preparatory-study-for-the-introduction-of-a-durability-index.html>
- [5] Zak Argabrite, Jim Murphy, Sally Jane Norman, and Dale Carnegie. 2022. Technology is Land: Strategies towards decolonisation of technology in art-making. In *NIME 2022*. PubPub, The University of Auckland, New Zealand. <https://doi.org/10.21428/92fbeb44.68f7c268>
- [6] Cornelis P. Baldé, Ruediger Kuehr, Tales Yamamoto, Rosie McDonald, Elena D'Angelo, Shahana Althaf, Garam Bel, Otmar Deubzer, Fernandez-Cubillo Elena, Vanessa Forti, Vanessa Gray, Sunil Herat, Shunichi Honda, Giulia Iattoni, Deepali S. Khetriwal, Vittoria Luda di Cortemiglia, Yuliya Lobuntsova, Innocent Nnorom, Noémie Pralat, and Michelle Wagner. 2024. *Global E-waste Monitor 2024*. Technical Report. International Telecommunication Union (ITU) and United Nations Institute for Training and Research (UNITAR). <https://research-repository.griffith.edu.au/items/400c890e-a99e-4426-84b7-df1e4b08902b>
- [7] Eli Blevins. 2007. Sustainable interaction design: invention & disposal, renewal & reuse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 503–512. <https://doi.org/10.1145/1240624.1240705>
- [8] Carlo Alberto Boano. 2021. Enabling Support of Legacy Devices for a more Sustainable Internet of Things: A position paper on the need to proactively avoid an "Internet of Trash". In *Proceedings of the Conference on Information Technology for Social Good*. ACM, Roma Italy, 97–102. <https://doi.org/10.1145/3462203.3475883>
- [9] Jean-François Charles, Carlos Cotallo Solares, Andrew Willette, and Carlos Toro Tobon. [n. d.]. Using the Axoloti Embedded Sound Processing Platform to Foster Experimentation and Creativity. ([n. d.]).
- [10] Brent Edstrom. 2016. *Arduino for Musicians: A Complete Guide to Arduino and Teensy Microcontrollers*. Oxford University Press. Google-Books-ID: WYLnCwAAQBAJ.
- [11] Valentin Girard, Maud Rio, and Romain Couillet. 2024. Computing, Complexity and Degrowth : Systemic Considerations for Digital De-escalation. (2024).
- [12] Aliénor Golvet, Benjamin Matuszewski, and Frederic Bevilacqua. 2024. Designing a distributed musical instrument for collective improvised interaction. In *Audio Mostly 2024 - Explorations in Sonic Cultures*. ACM, Milan Italy, 405–420. <https://doi.org/10.1145/3678299.3678341>
- [13] Craig Goodwin, Sandra Woolley, Ed de Quincey, and Tim Collins. 2023. Quantifying Device Usefulness - How Useful is an Obsolete Device?. In *Human-Computer Interaction - INTERACT 2023*, José Abdelnour Nocera, Marta Kristin Lárusdóttir, Helen Petrie, Antonio Piccinno, and Marco Winckler (Eds.). Springer Nature Switzerland, Cham, 90–99. https://doi.org/10.1007/978-3-031-42293-5_8
- [14] Gareth Halfacree and Ben Everard. 2021. *Get Started with MicroPython on Raspberry Pi Pico: The Official Raspberry Pi Pico Guide*. Raspberry Pi Press. Google-Books-ID: 954QEQAQBAJ.
- [15] Lon Ake Erni Johannes Hansson, Teresa Cerratto Pargman, and Daniel Sapiens Pargman. 2021. A Decade of Sustainable HCI: Connecting SHCI to the Sustainable Development Goals. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–19. <https://doi.org/10.1145/3411764.3445069>
- [16] Dylan A. Hazelwood and Michael G. Pecht. 2021. Life Extension of Electronic Products: A Case Study of Smartphones. *IEEE Access* 9 (2021), 144726–144739. <https://doi.org/10.1109/ACCESS.2021.3121733>
- [17] Garnet Hertz and Jussi Parikka. 2012. Zombie Media: Circuit Bending Media Archaeology into an Art Method. *Leonardo* 45, 5 (Oct. 2012), 424–430. https://doi.org/10.1162/LEON_a_00438
- [18] Steven J. Jackson and Laewoo Kang. 2014. Breakdown, obsolescence and reuse: HCI and the art of repair. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, Toronto Ontario Canada, 449–458. <https://doi.org/10.1145/2556288.2557332>
- [19] Théo Jourdan and Baptiste Caramiaux. 2023. Culture and politics of machine learning in nime: A preliminary qualitative inquiry. In *New interfaces for musical expression (NIME)*. <https://hal.science/hal-04075438/>
- [20] Noah Klugman, Meghan Clark, Pat Pannuto, and Prabal Dutta. 2018. Android Resists Liberation from Its Primary Use Case. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 849–851. <https://doi.org/10.1145/3241539.3267726>
- [21] Xun Li, Pablo J. Ortiz, Jeffrey Browne, Diana Franklin, John Y. Oliver, Roland Geyer, Yuanyuan Zhou, and Frederic T. Chong. 2010. Smartphone Evolution and Reuse: Establishing a More Sustainable Model. In *2010 39th International Conference on Parallel Processing Workshops*. IEEE, San Diego, CA, USA, 476–484. <https://doi.org/10.1109/ICPPW.2010.70>
- [22] Jasmine Lu. <https://orcid.org/0000-0002-1144-4016>, View Profile, Pedro Lopes, <https://orcid.org/0000-0001-6527-7084>, and View Profile. 2024. Unmaking Electronic Waste. *ACM Transactions on Computer-Human Interaction* 31, 6 (Dec. 2024),

- 1–30. <https://doi.org/10.1145/3674505> Publisher: Association for Computing Machinery.
- [23] L. Magnier and R. Mugge. 2022. Replaced too soon? An exploration of Western European consumers' replacement of electronic products. *Resources, Conservation and Recycling* 185 (Oct. 2022), 106448. <https://doi.org/10.1016/j.resconrec.2022.106448>
- [24] Tamar Makov and David Font Vivanco. 2018. Does the Circular Economy Grow the Pie? The Case of Rebound Effects From Smartphone Reuse. *Frontiers in Energy Research* 6 (May 2018). <https://doi.org/10.3389/fenrg.2018.00039> Publisher: Frontiers.
- [25] Aymeric Mansoux and Brendan Howell. 2023. Permacomputing Aesthetics: Potential and Limits of Constraints in Computational Art, Design and Culture. (2023).
- [26] Raul Masu, Fabio Morreale, and Alexander Refsum Jensenius. 2023. The O in NIME: Reflecting on the Importance of Reusing and Repurposing Old Musical Instruments. 106–115. <https://doi.org/10.5281/zenodo.11189120> ISSN: 2220-4806.
- [27] Benjamin Matuszewski. 2019. Soundworks - A Framework for Networked Music Systems on the Web - State of Affairs and New Developments. In *Proceedings of the Web Audio Conference (WAC) 2019*. Trondheim, Norway. <https://hal.science/hal-02387783>
- [28] Benjamin Matuszewski. 2020. A Web-Based Framework for Distributed Music System Research and Creation. *AES - Journal of the Audio Engineering Society Audio-Acoustics-Application* (Oct. 2020). <https://hal.science/hal-03033143> Publisher: Audio Engineering Society Inc.
- [29] Benjamin Matuszewski, Norbert Schnell, and Frederic Bevilacqua. 2019. Interaction Topologies in Mobile-Based Situated Networked Music Systems. *Wireless Communications and Mobile Computing* 2019, 1 (2019), 9142490. <https://doi.org/10.1155/2019/9142490> _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2019/9142490>.
- [30] Lea Moseoso, Nolwenn Maudet, Edlira Nano, Thomas Thibault, and Aurélien Tabard. 2023. Obsolescence Paths: Living with Aging Devices. In *2023 International Conference on ICT for Sustainability (ICT4S)*. IEEE, Rennes, France, 13–23. <https://doi.org/10.1109/ICT4S58814.2023.00011>
- [31] Ulrich Norbistrath, Perseverance Munga Ngoy, Renato Perotto Machado, Huber Flores, and Mohan Liyanage. 2025. Empowering Sustainability: Upcycling Smartphones as the Future of IoT and Edge Computing in Emerging Economies. In *Proceedings of 23rd International Conference on Informatics in Economy (IE 2024)*, Cristian Ciurea, Paul Pocatilu, and Florin Gheorghe Filip (Eds.). Springer Nature, Singapore, 203–213. https://doi.org/10.1007/978-981-96-0161-5_18
- [32] Nicolas Nova. 2018. *Figures mobiles: une anthropologie du smartphone*. Ph. D. Dissertation. Université de Genève, Geneva.
- [33] Nicolas Nova and Gauthier Roussilhe. 2020. Du low-tech numérique aux numériques situés. *Sciences du Design* n° 11, 1 (May 2020), 91–101. <https://doi.org/10.3917/sdd.011.0091>
- [34] Helena de Souza Nunes, Federico Visi, Lydia Helena Wohl Coelho, and Rodrigo Schramm. 2019. SIBILIM: A low-cost customizable wireless musical interface. 15–20. <https://doi.org/10.5281/zenodo.3672850> ISSN: 2220-4806.
- [35] Shailesh Prabhu N and Ritanjali Majhi. 2023. Disposal of obsolete mobile phones: A review on replacement, disposal methods, in-use lifespan, reuse and recycling. *Waste Management & Research: The Journal for a Sustainable Circular Economy* 41, 1 (Jan. 2023), 18–36. <https://doi.org/10.1177/0734242X221105429>
- [36] Christian Remy and Elaine May Huang. 2014. Addressing the obsolescence of end-user devices: Approaches from the field of sustainable HCI. *Advances in Intelligent Systems and Computing* 310 (2014), 257–267. https://doi.org/10.1007/978-3-319-09228-7_15 ISBN: 9783319092287 Number: 310 Place: Heidelberg/New York Publisher: Springer.
- [37] Mohammad Rashidujjaman Rifat, Hasan Mahmud Prottoy, and Syed Ishtiaque Ahmed. 2019. The Breaking Hand: Skills, Care, and Sufferings of the Hands of an Electronic Waste Worker in Bangladesh. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. <https://doi.org/10.1145/3290605.3300253>
- [38] Clara Rigaud, Gilles Bailly, Ignacio Avellino, and Yvonne Jansen. 2022. Exploring Capturing Approaches in Shared Fabrication Workshops: Current Practice and Opportunities. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), 391:1–391:33. <https://doi.org/10.1145/3555116>
- [39] Clara Rigaud, Gilles Bailly, and Yvonne Jansen. 2023. Ressources de connaissances dans les ateliers de fabrication : objectifs et défis. In *IHM '23: Proceedings of the 34th Conference on l'Interaction Humain-Machine*. ACM, Troyes. <https://doi.org/10.1145/3583961.3583974>
- [40] Curtis Roads. 1988. Introduction to Granular Synthesis. *Computer Music Journal* 12, 2 (1988), 11–13. <https://doi.org/10.2307/3679937> Publisher: The MIT Press.
- [41] Charles Roberts, Graham Wakefield, and Matthew Wright. 2017. 2013: The Web Browser as Synthesizer and Interface. In *A NIME Reader*, Alexander Refsum Jensenius and Michael J. Lyons (Eds.). Vol. 3. Springer International Publishing, Cham, 433–450. https://doi.org/10.1007/978-3-319-47214-0_28 Series Title: Current Research in Systematic Musicology.
- [42] David Roedl, Shaowen Bardzell, and Jeffrey Bardzell. 2015. Sustainable Making? Balancing Optimism and Criticism in HCI Discourse. *ACM Transactions on Computer-Human Interaction* 22, 3 (June 2015), 1–27. <https://doi.org/10.1145/2699742>
- [43] Jennifer Switzer, Gabriel Marcano, Ryan Kastner, and Pat Pannuto. 2023. Junkyard Computing: Repurposing Discarded Smartphones to Minimize Carbon. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS 2023)*. Association for Computing Machinery, New York, NY, USA, 400–412. <https://doi.org/10.1145/3575693.3575710>
- [44] Carla Tapparo, Brooke Chalmers, and Victor Zappi. 2023. Leveraging Android Phones to Democratize Low-level Audio Programming. 288–294. <https://doi.org/10.5281/zenodo.11189186> ISSN: 2220-4806.
- [45] Ron Wakkary, Audrey Desjardins, Sabrina Hauser, and Leah Maestri. 2013. A sustainable design fiction: Green practices. *ACM Transactions on Computer-Human Interaction* 20, 4 (Sept. 2013), 23:1–23:34. <https://doi.org/10.1145/2494265>
- [46] Garrath T. Wilson, Grace Smalley, James R. Suckling, Debra Lilley, Jacquetta Lee, and Richard Mawle. 2017. The hibernating mobile phone: Dead storage as a barrier to efficient electronic waste recovery. *Waste Management* 60 (Feb. 2017), 521–533. <https://doi.org/10.1016/j.wasman.2016.12.023>
- [47] Trevor Zink, Frank Maker, Roland Geyer, Rajeevan Amirtharajah, and Venkatesh Akella. 2014. Comparative life cycle assessment of smartphone reuse: repurposing vs. refurbishment. *The International Journal of Life Cycle Assessment* 19, 5 (May 2014), 1099–1109. <https://doi.org/10.1007/s11367-014-0720-7>

A Test resources

Listing all the USB serial devices on bash.

```
ls /dev/{tty,cu}.*
```

Obtaining information about an Android device from adb.

```
adb shell getprop ro.build.version.release
```

```
adb shell getprop ro.product.cpu.abi
```

Downloading necessary packages on the Android phone via Termux.

```
pkg update --allow-unauthenticated
pkg install nodejs-lts git openssl-tool
node -v #should give the node version
npm -v #should give the npm version
```

Downloading the test environment using git.

```
git clone --recursive [repository].git
```

[repository] is available at <https://zombitron.clararigaud.com>
under GPU GPL-3.0 licence.