Crafting Interactive Experiences with Non-programmers

Crafting Interactive Experiences

Chris Greenhalgh

Mixed Reality Lab, University of Nottingham, Chris.Greenhalgh@nottingham.ac.uk

The Mixed Reality Lab has a long history of creating public interactive experiences in collaboration with creative practitioners. Looking across four such experiences, this keynote explores the role of code (i.e., bespoke software) in making them possible, the practicalities of non-programmers "authoring" key parts of the experience, the relationship between coding and knowledge production, and the changing nature of technical responsibilities. As well as being personally and inherently satisfying, the practical realization of novel interactive systems manifests new creative "materials", which open the door to new experiences and understandings of people and the world.

CCS CONCEPTS • **Human-centered computing~Human computer interaction (HCI)~Interactive systems and tools** • Applied computing~Arts and humanities~Performing arts • Human-centered computing~Interaction design~Interaction design theory, concepts and paradigms

Additional Keywords and Phrases: authoring

ACM Reference Format:

Chris Greenhalgh. 2023. Crafting Interactive Experiences with Non-programmers. In Companion Proceedings of the 2023 ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '23 Companion). Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/3596454.3597174.

1 INTRODUCTION

As a child of the "home computer generation", programming has been a major part of my life for over 40 years. There is something viscerally satisfying about typing some characters, "saving" them and then watching them come to life as a running program. At one level, it depends on many other people's contributions: to the hardware, the operating system, the editor, the compiler and so on. But it also has a sense of *creatio ex nihilo* – creation from nothing: until I typed it, that particular program did not exist; now it does, and it may become useful to others in its own way. Sometimes I have to tell my PhD students that you can't get a PhD just for writing code. But there are also many PhDs that would never have existed without (a lot of) code being written. This keynote explores one particular area of endeavor – creating interactive experiences with *non*-programmers – in which, to abuse a nuclear physics metaphor, code collides with *non*-code to yield, at the very least, a range of interesting byproducts.

2 PORTFOLIO

I will draw on four distinctive and contrasting interactive experiences: *Day of the Figurines*, *The Rough Mile*, *Climb!* and *Losing Her Voice*, which are introduced below. They are all public experiences or performances of some form, and each

one has been co-developed by the Mixed Reality Lab together with non-programming creative practitioners. I will then trace some of the common threads and themes that run through these examples.

2.1 Day of the Figurines

Day of the Figurines is a massively multiplayer game, played using text messaging, developed with performance arts group Blast Theory, in collaboration with the Fraunhofer Institute and Sony Net Services as part of the iPerG European research project [1]. It premiered in Berlin in 2006, i.e. before the announcement of the iPhone, at a time when Nokia still dominated the mobile phone market. The experience was anchored in a public physical installation, the centerpiece of which was a large model of an imaginary town, made up of the silhouettes of key locations. Each player selected a small unique figurine to represent themselves within the game, which was placed on the town model. Every 30 minutes the artists would ceremoniously move the players' figurines to reflect their activities in the game. The game unfolded over 24 game hours, which were played at one-tenth real-time over 24 days (10 hours per day). Players would receive updates from the game by text message, and could reply with simple commands in the style of text-only roleplaying game.

2.2 The Rough Mile

The Rough Mile is an experimental GPS-driven location-based audio experience delivered via an Android smartphone, that follows a pre-defined walking route around part of Nottingham City Centre [2]. *The Rough Mile* has a complex layered soundscape which combines: ambient environmental sound recordings, multitrack backing music, commercial songs and spoken word recordings. The audio experience unfolds along the route, with musical transition triggered as the listener passes physical waypoints. The soundscape and background music were design and composed by an HCI researcher with a background in composition and another researcher with a background in performance wrote the script for the narrator. In its full form there were two linked audio experiences intended for a pair of friends: first one person would follow the path, along which they were prompted to choose appropriate music tracks; second (and some time later) the other person in the pair would follow the same path, but would hear the chosen music tracks integrated into the soundscape.

2.3 Climb!

Climb! is a non-linear classical work for Piano-Disklavier and electronics, composed by Maria Kallionpää, and developed by the Mixed Reality Lab with colleagues at the University of Oxford [3]. The work is based around the metaphor of the pianist climbing a mountain, and is broken down into 17 micro-compositions (effectively short movements), which are organized along three intersecting musical "paths". The performance system comprises: a digital score for the pianist to follow, sequenced Disklavier accompaniments for some fragments (so that the piano duets with the pianist), a large on-stage projection visualizing the journey and dynamic sound effects (e.g. wind and other "weather" effects). Embedded within the work are a number of key passages or "musical codes", if these are played "incorrectly" (in the system's view) then the path taken through the piece changes. The system also drives a simple mobile webapp, which allows audience members to follow the progress of the performance, and to compare the "route" taken by the current performance with other performances. In its premiere performance in 2017 the work was played twice in succession, each performance (deliberately) taking different routes through the piece.

2.4 Losing Her Voice

Losing Her Voice is an opera about the life of American opera singer and silent film star Geraldine Farrar, written by composer and academic Elizabeth Kelly. It premiered at Lakeside, Nottingham, in April 2019, with support from Lakeside

Arts and Nottingham Trent University [4]. Kelly was inspired by the audience app developed for the *Climb!* performance, and wanted to make audience interaction a core component of the opera (together with some on-stage mixed reality). During the initial (exploratory) research & development process Kelly and the technical team jointly settled on (faux) social media as a unifying metaphor for this audience interaction. The developed audience interaction system had three interfaces: an audience webapp, (linked) on-stage projections and a back-stage management interface. The final six months of development brought in the professional stage director and venue team, which included theatre lighting and digital projection specialists. During this time the precise timing and extent of the audience interaction was refined and agreed with the composer and director. The final experience allowed the audience to use their mobile phones at key points within the show (as well as before and after) to respond to (faux) social media posts and quizzes, and in the final scene to echo a recording of Geraldine from their phone.

3 DISCUSSION

I would like to use this opportunity to trace four thoughts or threads through this body of work: the nature of code, the practicalities of authoring, the production of knowledge, and the changing nature of interactive system development. I will consider each in turn.

3.1 What is Code Good For?

Each of these experiences was built on a substantial foundation of new or bespoke code. *Day of the Figurines* was built using the EQUIP2 middleware [5], with a tuple-space-like persistent Data Space forming its loosely coupled heart. *The Rough Mile* was developed for *daoplayer*, a bespoke Android application which was unusual among location-based audio apps in allowing sample-accurate synchronization and mixing of multiple audio tracks. The *Climb!* experience depended on a somewhat Baroque assemblage of the musicodes system (to handle the overall logic and musical codes), a digital score based on MELD, the Max/MSP visual programming environment plus the stage- and mobile-apps [6]. The interactive audience system in *Losing Her Voice*, as well as connecting audience devices to onstage projectors, was also integrated with QLab, a theatre automation system, which was used to control all of the diverse media elements of the live opera under the direction of the deputy stage manager.

In all of these experiences there was a dynamic and two-way relationship between the development of the code and the development of the experience. EQUIP2's loose data coupling allowed *Day of the Figurines* to employ quite disparate client interfaces in a holistic experience. Daoplayer's multi-track synchronization allowed *The Rough Mile* to use more sophisticated and dynamic soundscapes. *Climb!*'s non-linear game-like structure was, at least in part, a direct response to the technical possibilities created by the nascent musicodes system. The creative team for *Losing Her Voice* relied on the system's integration with more familiar and tested theatre technologies (including stage lighting, digital projection and pre-recorded sound) to give it a distinct role in the live performance.

Every program, every tool, has its own distinctive strengths and qualities – somewhat like the grain in a piece of timber. And, at least if the code is reliable, then those qualities become repeatable, explorable, learnable. Each new application should make new things possible for its users. In an ideal world they can take their time to explore and understand *what* it does, and need never know *how* it does it, becoming a new kind of creative "material" [7].

3.2 The Pragmatics of Creation

While bespoke code was foundational to each of these experiences there was also a deliberate movement in the development of every experience from coding to "authoring", which I will define as the *non*-programmer(s) in the project

being able to define and refine the final experience without asking a programmer to write code. This transition to authoring allows those who carry the greatest creative responsibility to take control and ownership of much of the final interactive experience. Practically, it also allows them to explore possibilities independently and much more quickly than would otherwise be possible.

In these projects the range of possible elements and interactions is largely defined by the code. The code also determines what other form(s) of data can influence the final interaction and in what ways. Thus the specific nature and scope of the system's support for authoring is decided and refined as part of the development of the system as a whole. Contrasted with a general purpose programming language this can be seen as a partial specialization of the system, or a degree of commitment to specific form(s) of behavior. One can also think of it as defining a specific "sub-genre" or "idiom" of experience, within which the non-programmer(s) can take control. The nature and scope of this freedom or control is a key aspect of the "material" qualities of the system.

In the experiences considered above the potential to author is manifest variously as: simple predefined spreadsheets (*Day of the Figurines*), web forms (*Day of the Figurines* and *Climb!*), textual regular expressions (*Climb!*), specific JSON data file formats (*The Rough Mile*, *Losing Her Voice*) and mix-and-match fragments of JavaScript (*The Rough Mile*). We echo Dix et al. [8] in celebrating the potential of the humble spreadsheet; for example, later mobile listening projects adopted a multi-sheet spreadsheet format for authoring, which was translated into the JSON and JavaScript required by daoplayer. Our creative partners have often started with textual narratives or concept sketches. However the transition to even a basic predefined spreadsheet forces a degree of rigor and consistency that is both essential to the technical realization of the experience and also achievable by most of the people we have worked with (albeit sometimes challenging!).

3.3 From Code to Knowledge

In some practice-based fields the product is the main contribution; for example, in composition the body of music composed forms the bulk of a PhD submission. As I have already noted, however, that is not the case in Computer Science: a more abstract contribution is required. Reflecting on a range of the Mixed Reality Lab's artistic collaborations, we have previously argued the case for "Research-in-the-Wild" as a valid and valuable method in Human-Computer Interaction research [9]. From collaborations such as those presented here, we can trace virtuous connections between practice (i.e. artist-led experiences), studies (i.e. understanding experiences "in the wild") and theory (especially concepts and frameworks).

In this context it is the code that makes the practice possible. And it is often the new and unique features and qualities of the software that allow those experiences to innovate. And, in turn, it is often the novelty of the experience that reveals fresh insights into human behavior, interaction design and the like. In this mode, the pathway from code to knowledge can be somewhat convoluted. In order to make the intellectual payout that is demanded of them, those responsible for the code must follow these connections, and it important that they are invited and allowed to do so. This in turn depends almost entirely on a foundation of mutual (inter-disciplinary) respect.

It is worth noting that there is a second, more technical, pathway from code to knowledge, which is essentially one of abstraction and measurement. At the risk of being reductionist, the reader wishes to know, as a minimum: what is the general idea or principle that I could use in my own work; and what evidence can you give me that there is at least some situation in which it is a good idea, and perhaps even the best idea currently available for some rather specific situation (this was the path taken by [5]).

3.4 From Code to Interaction as a Service?

On a final note, we live in an increasingly connected world of virtualization, online services and ephemeral apps. Code is seldom shrink-wrapped, and Continuous Integration/Continuous Deployment practices and containerization can push new code to global use in minutes. All of the synchronous multi-user experiences discussed here employed loosely coupled distributed systems connected via web technologies (HTTP). The more recent ones (*Climb!* and *Losing Her Voice*) exploit Docker-based containers for development and deployment. Much as programmers are being integrated into DevOps teams, those of us who are engineering interactive systems increasingly find ourselves responsible not only for writing code, but also for the integration, deployment and operation of complex interactive networked services.

4 CONCLUSIONS

Amidst the varying demands of academic life, I sometimes feel a little envious of colleagues whose research methods seem to allow them to "just think" about things that may never exist, or walk up to things already made and simply observe them being used (or not). They seem to avoid all of the "messiness" and labor of actually making the thing and making it work, to a deadline and for a potentially paying public. And I might pause for a moment and reflect. And then to the "thinker" I might say: "the devil is in the detail", and it is only by actually bringing something into being that you can truly begin to appreciate its nature and qualities. And until other people actually encounter and experience it, neither we nor they can be sure how they will respond to it. And to the "observer" I might say: "come and see this; I think you will find it interesting". But I suspect that some of that is also a rationalization for the joy of making things work, and for giving life to other peoples' creative visions.

ACKNOWLEDGMENTS

This work was supported by UKRI through EPSRC grant [EP/T022493/1] Horizon: Trusted Data-Driven Products.

REFERENCES

[1] Steve Benford, Andy Crabtree, Martin Flintham, Chris Greenhalgh, Boriana Koleva, Matt Adams, Nick Tandavanitj, Ju Row Farr, Gabriella Giannachi, and Irma Lindt. 2011. Creating the spectacle: Designing interactional trajectories through spectator interfaces. ACM Trans. Comput.-Hum. Interact. 18, 3, Article 11 (July 2011), 28 pages. https://doi.org/10.1145/1993060.1993061

[2] Adrian Hazzard, Jocelyn Spence, Chris Greenhalgh, and Sean McGrath. 2017. The Rough Mile: Reframing Location Through Locative Audio. In Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences (AM '17). Association for Computing Machinery, New York, NY, USA, Article 41, 1–8. https://doi.org/10.1145/3123514.3123540

[3] Steve Benford, Chris Greenhalgh, Adrian Hazzard, Alan Chamberlain, Maria Kallionpää, David M. Weigl, Kevin R. Page, and Mengdie Lin. 2018. Designing the Audience Journey through Repeated Experiences. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). Association for Computing Machinery, New York, NY, USA, Paper 568, 1–12. https://doi.org/10.1145/3173574.3174142

[4] Chris Greenhalgh, Adrian Hazzard, Steve Benford, Laurence Cliffe, and Elizabeth Kelly. 2022. Crafting Trajectories of Smart Phone Use at the Opera. ACM Trans. Comput.-Hum. Interact. 29, 6, Article 59 (November 2022), 37 pages. https://doi.org/10.1145/3531007

[5] Chris Greenhalgh, Steve Benford, Adam Drozd, Martin Flintham, Alastair Hampshire, Leif Oppermannm Keir Smith, Christoph von Tycowicz. 2007. Addressing Mobile Phone Diversity in Ubicomp Experience Development. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds) UbiComp 2007: Ubiquitous Computing. UbiComp 2007. Lecture Notes in Computer Science, vol 4717. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-74853-3_26

[6] Maria Kallionpää, Chris Greenhalgh, Adrian Hazzard, David M. Weigl, Kevin R. Page, Steve Benford. 2017. Composing and realising a game-like performance for disklavier and electronics. Published in: NIME 2017: New Interfaces for Musical Expression, Copenhagen, 15-18 May 2017 : proceedings.

[7] Andrew Bluff and Andrew Johnston. 2019. Devising Interactive Theatre: Trajectories of Production with Complex Bespoke Technologies. In Proceedings of the 2019 on Designing Interactive Systems Conference (DIS '19). Association for Computing Machinery, New York, NY, USA, 279–289. https://doi.org/10.1145/3322276.3322313

[8] Alan Dix, Rachel Cowgill, Christina Bashford, Simon McVeigh, and Rupert Ridgewell. 2016. Spreadsheets as User Interfaces. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '16). Association for Computing Machinery, New York, NY, USA, 192–195. https://doi.org/10.1145/2909132.2909271

[9] Steve Benford, Chris Greenhalgh, Andy Crabtree, Martin Flintham, Brendan Walker, Joe Marshall, Boriana Koleva, Stefan Rennick Egglestone, Gabriella Giannachi, Matt Adams, Nick Tandavanitj, and Ju Row Farr. 2013. Performance-Led Research in the Wild. ACM Trans. Comput.-Hum. Interact. 20, 3, Article 14 (July 2013), 22 pages. https://doi.org/10.1145/2491500.2491502