

Software Literacy and Creative Industries

Greg Giannis¹, Hugh Davies², Victoria Moulder³, Murray Mckeich⁴

¹Independant Researcher, Australia, giannis.greg@gmail.com, ²La Trobe University, Australia, h.davies@latrobe.edu.au, ³Simon Fraser University, Canada, vmoulder@sfu.ca, ⁴RMIT University, Australia, murray.mckeich@rmit.edu.au

Abstract

We are living in a world where software is central to every field of social, political, and economic import. Governments are imposing coding into primary and secondary school curricula as they begin to understand the importance and urgency of engaging with these technologies. How do we make sense of and trust these myriad manifestations of software, or even know or understand who or what is behind the code that creates and designs our mediated reality? How can we have agency to disrupt and change mainstream society's dominant control through this meta-medium? Where do we start? This panel will share our knowledge of strategies to promote critical engagement with software.

Introduction

This Panel session will consist of four 15-minute presentations as described in the sections to follow. Panelists will address the following questions:

-Artists and Code. What strategies are used to encourage artists to utilise coding in their practice?

-Computational Thinking in Art/Design Education – How can we develop Computational Thinking as a fundamental foundation of creative imagination?

-Transcoding Place through media platforms, will answer, what considerations do designer's have in the production of artscience collaborations and why are they important?

-A revolution in coding is required within universities What is needed to improve academia's software literacy?

Greg Giannis: Artists and Code

I am undertaking research to investigate examples of best practice for the teaching of software and hardware skills to students in the creative industries. I have been exploring this in my classes during the last 10 years or so with digital media and visual art students.

The teaching of coding has become a somewhat topical area given the recent introduction of coding to Australian primary and secondary school curricula, and the emphasis on STEM to STEAM¹. Most of this discussion revolves around how to improve engagement with STEM and I (and others) argue, and now there is evidence to support this [1],

that engagement can be facilitated by tapping into students' creativity.

Whilst there are many examples and anecdotal evidence of student engagement in STEM through art, there is limited evidence of art students engaging with STEM. The art science nexus is pertinent and organisations such as the Australian Network for Art & Technology in Australia, the Art and Technology Program of the Los Angeles County Museum of Art, (1967-1971) [2] and NYU's ITP Masters program have made great contributions in bridging this divide. I am arguing that it is imperative that there is engagement with these technologies, in particular software, given its pervasiveness, otherwise we limit any form of resistance against the society of control. Fuller's field of software studies [3] is pertinent. Lovink [4] also argues that in order to challenge the society of control we need to break down the barriers between the humanities and hard sciences so that there can be some form of critical engagement with the hard sciences. How can we encourage artists to be part of this discussion? If we accept that artists play a critical role in society then there needs to be engagement with these technologies, and learning to code could be a starting point. I know that there are many artists already doing this, and many art courses do have elements of coding in their courses, but these tend to be exceptions driven by progressive teachers and artists.

Hugh Davies: Computational Thinking in Art/Design Education

Over the last two decades, the evolution of art and design curriculum has been significantly impacted by the challenges and potential of digital disruption in the creative industries. Creative software tools first emulated then largely replaced manual production practices. There is growing evidence that another, and possibly greater, paradigm shift is eminent.

Predictions for the growth of Artificial Intelligence and its possible impact on employment are currently an issue of growing social concern. The current emergence of Procedural Content Generation software in the games industry, online journalism, architecture and music industry suggest that commercial creative practices will not be immune to the future impact of creative A. I.

¹ STEM to STEAM (<http://stemtosteam.org>) is a movement to include the arts and design into science based programs so that innovation is influenced by the humanities.

Possible scenarios for the future may see practitioners eschew today's generic software tools for bespoke applications that operate as semi-autonomous studio assistants. This has the potential to be a positive development that will offer new creative opportunities for those that embrace it.

The education of artists and designers for this environment may require some level of programming ability but of greater importance will be the development of Computational Thinking as a fundamental foundation of creative imagination. Computational Thinking defines a common conceptual space that will afford access to creative partnerships between human cognition and A.I.

Victoria Moulder: Transcoding Place Through Digital Media

Over the last ten years, alternate reality and location-based technologies have rapidly transformed in response to the adoption of mobile and social computing [5]. In tandem with the rise of these technologies is an ever-increasing community of people with 21st century media literacy skills for deconstructing and reconstructing narratives that have the potential to equally influence the flow of mainstream media.

For this panel section I will discuss the findings of a multiple case study comparing the practices used by designers to produce alternate reality games. I report how these methods facilitate cross-discipline collaborations. In addition, how designers work with non-profit organizations to build awareness campaigns that support larger philosophical goals and future thinking possibilities.

Murray Mckeich: Universities Required a Coding Revolution

Although workers in academia are fluent in digital technologies at an email, database, spread-sheet level, they remain largely illiterate to languages of code, and to the broader conceptual aspects of the networked world. Outside of studies in computer science, matters of code, the influence of computational thinking and the extent to which algorithms shape the perceptions and conditions of reality, remain largely untaught. Technological literacies that are now central to every field of social, political, and economic import are being ignored by the academy. As a result, universities that have been long celebrated as institutions to instruct and celebrate literacy are themselves becoming increasingly illiterate. Languages of code, and their broader implications need to be taught in universities, but prior that, they must be taught to the universities. Beyond academic buzz terms of future-readiness and innovation, in order to remain fundamentally relevant and rhetorically coherent, the academy must catch-up to the present.

References

[1] J. M. Eger, "Arts Based Learning of STEM Works Says NSF Funded Research Firm" website. Accessed December

28, 2015 http://www.huffingtonpost.com/john-m-eger/arts-based-learning-of-st_b_8724148.html

[2] LACMA 50 website. Accessed December 28, 2015, <https://archive.org/details/reportonarttechn00los>

[3] M. Fuller, (Ed) "Software Studies: A Lexicon" MIT Press, 2008.

[4] Papastergiadis & Lynn, (Eds) "Art in the Global Present" p 107. UTS ePRESS, 2014.

[5] E. Bonsignore, V. Moulder, C. Neustaedter, D. Hansen, K. Kraus, and A. Druin "Design tactics for authentic interactive fiction" pp. 947-950. CHI ACM, 2014.