International Journal of Child-Computer Interaction ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

journal homepage: www.elsevier.com/locate/ijcci

Playing Beowulf: Bridging computational thinking, arts and literature through game-making

Bruno Henrique de Paula [a,*], Andrew Burn [a,*], Richard Noss [a], José Armando Valente [b]

[a] UCL Knowledge Lab, UCL Institute of Education, University College London, London, UK
[b] Instituto de Artes—Universidade Estadual de Campinas, Campinas, Brazil

ARTICLE INFO

Article history:
Received 3 February 2017
Received in revised form 2 November 2017
Accepted 9 November 2017
Available online xxxx

Keywords:
Game-making
Computational thinking
Arts
Humanities

ABSTRACT

Preparing younger generations to engage meaningfully with digital technology is often seen as one of the goals of 21st century education. Jeanette Wing's seminal work on Computational Thinking (CT) is an important landmark for this goal: CT represents a way of thinking, a set of problem-solving skills which can be valuable when interacting with digital technologies, and with different fields of knowledge, such as Arts and Humanities. Even if this cross-areas relevance has been celebrated and acknowledged in theoretical research, there has been a lack of practical projects exploring these links between CT and non-STEM fields. This research develops these links. We present a specific case – a game produced by two 14 years-old boys – within *Playing Beowulf*, a collaboration with the British library's Young Researchers programme, in which students aged 13–14 from an inner-London (UK) school have developed games based on their own readings of the Anglo-Saxon poem Beowulf during an after-school club. The game was produced using *MissionMaker*, a software (currently under development at UCL Knowledge Lab) that allows users to create and code their own first-person 3D games in a simple way, using pre-made 3D assets, such as rooms, props, characters and weapons and a simplified programming language manipulated through drop-down lists. We argue that *MissionMaker*, by simplifying the development process (low floor), can be a means to foster the building of knowledge in both STEM (CT) and Arts and Humanities, building bridges between these two areas inside and outside traditional schooling.

1. Introduction

In recent decades, we have witnessed a great dissemination of the benefits of computer-related skills for Education. Notions such as *procedural thinking* [1], *procedural literacy* [2,3] or *computational literacy* [4] have fed into a valorisation of computational skills in schools in different countries. Jeanette Wing's [5] work is often recognised as one of the landmarks of this new movement: she envisaged her term "Computational Thinking" (CT) as a "formative skill" for the contemporary world, at least as important as literacy and numeracy [6], essential for citizens' lives in an era in which digital technologies are ubiquitous.

The importance of this "formative skill" is clear in different domains. In media, for instance, we can directly relate it to what Manovich [7] names "new media" (digital media). These are constituted by two different layers: the 'cultural layer', which refers to cultural forms – the "language" – of media (e.g. film, television, videogames and, recently, the 'convergence culture' [8]), and the 'computer layer', referring to the information-processing aspects of "new media" [7], which modify how the "language" of media functions. In "new media", meaning emerges from this confluence between these two layers.

However, even if the cultural and computer layers are inextricably entangled, conceptions of media literacy in recent years [9,10] have usually focused on the cultural aspect. By contrast, conceptions of computational thinking have focused on the computer aspect, not necessarily emphasising it as a social practice with deep cultural and communicational impacts [11]. Therefore, we need to re-examine how people think about culture and computing, both in relation to the (media) arts, and in relation to computing.

In this paper, we focus in how CT, as a "formative skill", can help to establish bonds between computing-related concepts and the conceptual frameworks required by the media and the arts. We focus on a specific approach: digital game-making. We present a case study from a project in which a different perspective towards game-making and formal learning was employed: by using the Anglo-Saxon poem *Beowulf* as the basis for the project, we aimed at understanding how CT and Arts and Humanities (AH) can be fostered simultaneously in an educational context.

The main results and further discussions related to this case study will be presented in this paper in Section 4. Prior to that,

* Corresponding authors.
E-mail addresses: bruno.paula.15@ucl.ac.uk (B.H. de Paula), a.burn@ucl.ac.uk (A. Burn).

https://doi.org/10.1016/j.ijcci.2017.11.003
2212-8689/

Section 2 introduces a brief literature review on relevant aspects regarding CT and on the theoretical foundations that guided our work on bridging CT and AH. Additionally, Section 3 describes the tool employed in our project – *MissionMaker* – as well as the main methods used to structure and analyse our data.

2. Context

Jeanette Wing's works [5,12] are often seen as landmarks for a new phase on computers and computer-related skills in schools. Since then, CT is emerging as a desired learning goal in different educational systems, even becoming part of official educational policies in some countries [13,14]. This does not mean, however, that CT can be considered a stable, closed term [15], and several definitions have emerged trying to pin down its core concepts [5,12,16]. Nevertheless, there is some consensus about foundational aspects such as abstraction, automation and reduction; and, more importantly, that computational thinking refers to a set of problem-solving skills based on computer-related concepts [17].

But how have different countries incorporated CT into educational systems? One of the current trends among policy-makers is to use activities based on coding and programming to foster CT. Some of these initiatives include "Computer Science is for everyone" [18], an effort from the US White House to promote and popularise Computer Science (CS), and the new National Curriculum in England, which has CS as one of its bases [13].

While CT can certainly be fostered through this path, and some authors [19] argue that programming is essential to engage with CT's common practices, CT cannot be understood as a synonym – or even an "abridged version" – of CS. While the latter is recognised as an academic discipline that studies computers and computational systems, the former is understood as a specific set of thought processes used to solve complex problems [20]. In this sense, CT is not – and should not be taught as if it were – necessarily bound to CS, since it can be used as a general tool in different domains and not only in those that are part of CS:

> the ultimate goal [of CT] is not to teach everyone to think like a computer scientist, but rather to teach them to apply these common elements to solve problems and discover new questions that can be explored within and across all disciplines [21].

Some initiatives aimed to foster CT in this broader sense, exploring its alleged cross-curricular potential through different scenarios, such as games, journalism, science and engineering [22]. Other research also presented an array of activities with potential to embody this broader view on CT, such as the Computer Science Unplugged, where students learned computational concepts without relying on digital technologies, or initiatives exploring robotics, digital narratives, simulations and games [23–25].

Producing games is arguably one of the most consolidated approaches to deliver CT in educational contexts [21,26], mainly because it provides an opportunity to connect several "formative skills" within the same task, while also stimulating students' creativity and engagement with diverse modes of communicating meanings. Another argument in favour of games when aiming at CT is their systemic nature [27]: games are often condemned and considered frivolous [28], but we cannot ignore that, in many cases, they are some of the first experiences that we have of formal systems, fostering the understanding of correlations between rules and outcomes and of how to operate these rules.

However, even if this possibility of cross-curricular work through games is celebrated throughout the literature [29,30], the majority of practical work in this field is still bound to STEM-related academic content or to the sole development of programming principles [26], while there is a shortage of initiatives trying to establish stronger bonds with other parts of the curriculum, especially with AH.

One path to bridge this gap and connect CT and AH is through Proceduralism [31–33]. Murray [33] argues that one of the essential properties of digital environments is procedurality, which allows them to execute a series of rules and, consequently, opens an opportunity for "encapsulating specific real-world behaviours into programmatic representations" [31, p. 13]. In other terms, this capacity of creating loops and specific logic-oriented sequences modifies how we can use media to express meaning.

This capacity is further explored by Bogost [32] under the scope of procedural rhetoric: the idea that arguments can be established and used to persuade an audience through processes in any kind of cultural production: "any medium—poetic, literary, cinematic, computational—can be read as a configurative system, an arrangement of discrete, interlocking units of expressive meaning" [32, p. ix].

Procedural rhetoric sees in processes (especially computational ones, such as game rules) the ability to frame situations and to express meanings to an audience (players, spectators, readers). It signals the importance of being able to read systems and digital processes not only in a functional way (e.g. how to give specific instructions to a computer to make a character jump in a game), but also to understand what arguments might be implicit in these processes (e.g. what does it mean to give the ability to the character to jump in a game? Can the character run and jump at the same time? And what is the narrative significance of jumping?) [32]. Examples of this application can range from this silly "jumping" routine to more complex (and maybe, more controversial) ones, such as an hypothetical online RPG games in which avatars have genders and can have affective relationships among themselves, but only if they are from different genders. In this sense, it is clear that the game – through its code – is giving a powerful message about same-sex relationships.

Proceduralism, in some sense, highlights the importance of thinking about how processes (computational or not) can be interconnected and communicate meaning; this comprehension can be seen as one of the first steps for the problem-solving heuristic defended by CT, and are part of what Bogost defines as *procedural literacy* – "[...] the ability to reconfigure basic concepts and rules to understand and solve problems, not just on the computer, but in general" [2, p. 32].

Proceduralism as a current of thought is often criticised for some of its positioning: Sicart, for instance, considers that it favours the designer/writer's ideas rather than acknowledging the value of other possible personal interpretations by the audience [34]. While this critique is relevant – and we should bear it in mind while approaching works from a proceduralist perspective – it emphasises how computer-related processes are not necessarily neutral, but can carry values and communicate meanings in different ways. The proceduralist argument reminds us that, after all, CS and AH are not so far apart as they might seem at first sight: meaning is produced in all cases, and although the meaning carried by a narrative might seem more evident in a first moment, code and computational processes also generate meaning and demand interpretation; thus, when working within this domain we should be aware of what is being communicated by our production. Proceduralism, however, is not the only way to approach CS and AH.

Other research successfully explored narratives as a means to help students deal with CT-related concepts, such as abstraction. Here, we understand abstraction as a kind of generalisation by removing detail from a complex object or process, in order to construct general concepts that might be used in other objects/processes [35]. Mathematics, due to its nature, has in abstraction one of its core elements and often pupils struggle to move from its abstract forms to more concrete ones. Mor and