

# Why Learning Creative Code Still Matters in the Age of Artificial Intelligence

By Kieran Christopher Monteiro

## Introduction

**A Changing Landscape** - Artificial Intelligence (AI) has had a significant influence across numerous sectors achieving new milestones for humanity. ChatGPT, a chatbot developed by OpenAI, gained significant traction due to its advanced code-writing capabilities and natural language processing (OpenAI, 2022). This development raises the question of whether AI's stronghold in the programming landscape will undermine artists making code-based art. And, amid this surge of technological progress, one could argue that human ingenuity risks being stifled if we completely surrender the art of creative code to AI.

## Objective

To envision where a line is drawn between AI's intervention and traditional coding, we first need to look at what is lost for human ingenuity if the learning code traditionally for art is replaced purely by written prompts for AI to produce said code.

This paper, therefore, argues that learning code to make art is crucial for the following key reasons:

- 1) Learning code traditionally enhances **human creativity**.
- 2) Coding knowledge allows one to extend and refine the **creative process** beyond AI's sole output.
- 3) Learning creative coding promotes **computational thinking**.

## Human Creativity

Learning creative coding has shown links to improvements in creativity. Research papers such as by Şendağ, Yakin, and Gedik (2023) and Su et al. (2024) indicate an increase in Creative Thinking skills during both the explicit teaching of code and lessons on creative programming. This skill is important as it is defined as our ability to conceive original ideas, draw from personal experiences, and infuse creations with emotional and cultural significance (Runco & Jaeger, 2012)

## Creative Process

One could say that learning code extends the creative process beyond AI's initial output. Jonsson and Tholander's (2022) study indicated the inconsistent and imperfect behaviours of the AI generating programming code (i.e., the output commonly did not correspond to the participants' initial expectations or intentions) with participants relying on their code knowledge to refine their designs. This paper, therefore, suggests a union between AI and humans to extend and refine the artist's intent.

## Computational Thinking

Creative coding enhances Computational Thinking (CT) as stated by Romero et al. (2017) who found development in CT competency in learners though creative programming activities within the Zone of Proximal Creativity (ZPC). Aranda and Ferguson (2022) also discussed the integration of creative thinking within CT frameworks, emphasising that creative coding activities can enhance students' ability to think computationally. As such, creative coding acts as a bridge to acquiring CT which can be used in other domains.

## Creative Coding in Partnership with AI

This paper then turns to Sharples (2023) who devised a table outlining the roles that AI can possess that could be pertinent in how we view AI in the creative coding process; (1) A Possibility Engine – AI generates alternative ways of expressing art through code structures and snippets, (2) Collaboration Coach – AI helps the artist research and solve code problems in his written code, (3) Co-Designer – AI assists throughout the artmaking. With what was mentioned, and thinking about the next step for creative coding, I call upon an extension of John Maeda's view of Creative Coding, the incorporation of hardware as seen in Jan Zuiderveld's *Conversations Beyond the Ordinary*, the positioning of plants as artists in my artwork *Whatever Will Be, Will Be*, in recognition of Mills et al.'s (2024) discussion on the use of coding and CT to solve cross-disciplinary problems, to come up with a tetradic collaboration diagram as a next possible step for contemporary creative coding

