**SESSION O-2F**

**SOCIETAL IMPACTS OF EDUCATION AND LANGUAGE**

*Session Moderator: Judith A Howard, Sociology*

**MGH 284**

1:30 PM to 3:00 PM

*Note: Titles in order of presentation.*

**Forking a Sketch: How the OpenProcessing Community Uses Remixing to Collect, Annotate, Tune, and Extend Creative Code**

*Shenna Shim, Senior, Human Centered Design & Engineering*

*Mentor: Nadya Peek, Human-Centered Design and Engineering*

*Mentor: Blair Subbaraman, Human Centered Design & Engineering*

Creative coding refers to a method of computer programming that prioritizes artistic expression. Remixing is the iterative behavior of altering an existing artifact. Frameworks such as p5.js support sketching with creative code. Given the focus on expressivity over functionality, code reuse in creative coding practice is distinct from other programming contexts. Additionally, remixing facilitates iteration on existing code, but we have yet understand how creative coders use remixing in practice. To understand creative coder remixing strategies, we studied the community of OpenProcessing, a site dedicated to sharing code-generated artworks. To begin, we conducted a network analysis to determine which datasets of original sketches (also referred to as the antecedent sketch) and their remixes to use in our study. Our visualization consisted of a social network graph in which the nodes represent individuals, and edges showing their relationships. We found that 30% of the 1.2 million sketches in our dataset were involved in remixing. For data analysis, we utilized a code-diff tool to showcase ways the antecedent sketch’s code differs from the remix and categorize various types of remixing strategies. Over time, these categories became increasingly focused on changes made visible on our code-diff tool. We present on the diversity of ways that authors remix to curate projects, annotate process, explore variations, and transform existing sketches. Through remixing, artists have already begun to tailor, customize, and explore different ways to use their creative tools, in ways system developers may not have foreseen. We find that remixing also encourages exploratory programming and experiential learning. As creative code is increasingly used to support computational education, we can consider the implications of remixing for understanding and facilitating informal learning. At last, we reflect on the prevalence of these remix types and how future systems could support a multiplicity of remixing strategies for creative work.