The Design Game Project

Introduction

In the next six weeks we will inventing a new kind of software called a design game. Since this is a new kind of software it will be up to us to build a description of it. This process of definition will be carried out by building, and refining, the software itself.

We can probably say a few things, however, about what a design game might be. While design software is used as a means towards an end, a design game should be played. Unlike a traditional video game it should produce – whether a finished design or the seeds of a design. More specifically for the purposes of this course, a design game should be simple, engaging, and instrumental. It should be simple in it's design: focused in scope and easy to use. It should be engaging in its play. It should be instrumental by producing concrete products that can be transfered to further design processes.

The Process

Producing the software will follow a collaborative and iterative process. In teams of two or three, students will first sketch out their idea for a design game. At the same time we will be learning the basics of how to write code in processing, which will help us refine our ideas to realizable plans. After the sketch phase, an iterative cycle of code writing and editing will proceed, leading ultimately to the final design game prototype. Through this process, the instructor will act as an ad hoc member of each team suggesting resources and providing coding assistance for more complicated problems. At the end of each week long iteration, teams will be evaluated on the thoroughness of their exploration and the refinement of their design game in terms of simplicity, engagement, and instrumentality. Team members should keep records of their contributions, whether these are in the form of research, critique, coding, or other types of work. Evaluation of each teams' progress will be based on this work and not the quality or quantity of code written. Coding and research that yields failure will be judged as favorably as that which yields success, provided that team members can demonstrate what was learned in the process.

Tools

We will be producing our software with Processing, an open-source and free software development environment. Processing was written by Casey Reas and Ben Fry while they were students in the Media Lab at MIT. It was created specifically as an easy-to-learn programming environment for artists and designers. Because of this focus, it is especially well suited for creating graphic effects in both two and three dimensions. There is a large and active group of developers constantly adding and refining the functionality of the Processing environment. Projects developed using Processing typically depend heavily on this collaborative community, borrowing code and techniques liberally. We will be taking advantage of this situation, sharing not only among ourselves, but also with the larger Processing community.