

Critiquing Software Interactions

Applying Critical Design to Software

Lindsay Grace
Armstrong Professor of Interactive Media Studies
Miami University, School of Fine Arts



Software is philosophical.

Software is designed by people who have been influenced by a specific understanding of the way objects, people and systems work. These concepts are then transferred to the user, who manipulates that software, operating within the prescribed parameters.

The designer reinforces an understanding of the world that is emphasized by the software they use. The designer and user then produce works that mimic these same philosophies instead of departing from them.

Software Philosophy: 2009

- An analysis of the underlying
- themes and philosophies integrated software
- Diagnosing how the design of existing systems effects the design of new systems
- Expose the qualities of software interactions, software interoperability, and programming conventions that may effect the problem solving process



- Critical Gameplay:
 - Software Philosophy for **Game Design** and production
 - Begins with application of **Critical Design**
 - **Diagnose key questions** in how games are played
 - **Create games** that illustrate alternate ways to play



