Critiquing Software Interactions
Exposing the invisible effects of software on the problem solving processes

Lindsay Grace
Armstrong Institute for Interactive Media Studies
Miami University
ACH: Panel Presentation

Key Philosophies of Software
• The Heavy Use of Analogy
• The abundant Application of Reductivism
• An Emphasis on Transferred Agency

Critical Gameplay:
• Software Studies: 2009
  — Lev Manovic at UCSD
• Software Philosophy: 2009
  — An analysis of the underlying themes and philosophies integrated into 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
    • Exhibited in Europe (Greece), South America (Brazil) and North American (various) in 2009-2010

Critical Gameplay:
How do game mechanics effect the way we problem solve, socialize, or even view the world?

Critical Gameplay
• When we shoot do we learn to destroy obstacles instead of working around them?
Critical Gameplay
Does the **binary world** of 
**enemies** and **adversaries**
teach us to **ignore the**
gray in the everyday?

Iterative Design and Development

- Investigating these practices yields a 
  fundamental evaluation of the design process
  - IDEO Design Thinking
  - Iterative design
  - Collective, multidisciplinary practitioners
    - Global Game Jam

- Can we invert gameplay mechanics to better Harness Human-Computation in games?
- Can iterative processes like Design thinking and prototype thinking yield better HCI?

Critical Gameplay
Are we forgetting how to **play with** 
each other, because **playing**
against each other is 
more common?