Welcomes

UC Santa Cruz welcome

Noah, Michael, Chaim, Robin, Jane, Irena, Kate, Mike, Eric, April, & Bryan

NEA welcome

Joan & Bill

NEH
welcome

Brett & Jason

NSF
welcome
Janet

Microsoft welcome

Kent & Donald

Pamela welcome

Original NSF program officer

Workshop introduction & structure

Why are we here?

Conversation

Why are **we** here?

Good question

What unites all this work?

digital humanities

software & platform studies

interactive entertainment

game creation

educational innovation

interactive installations

transmedia narrative

design augmentation

reimagining scholarly publishing

procedural animation

cultural probes

art history & visual studies

digital art

computer science

feature film production

artificial intelligence

Why do connections seem to be forming?

- NSF/NEA "Re/Search" convening

- NSF/NEH "Digging Into Data" challenge

- Microsoft Kinect collaborations with universities and hackers

- MIT Press software/ platform studies

- MacArthur Digital Media and Learning

- NEH forms Office of Digital Humanities

- NEA Media Arts includes games

- NSEAD & XSEAD initiatives

- much more!

Seeing computational processes as part of culture

We need each other to move forward

What will we do here?

What will we do here?

- Discuss five themes

- Develop white paper priorities

- Develop multi-institution project ideas

- Have a conversation with field builders

- Select, refine, and develop project plans

Theme 1:
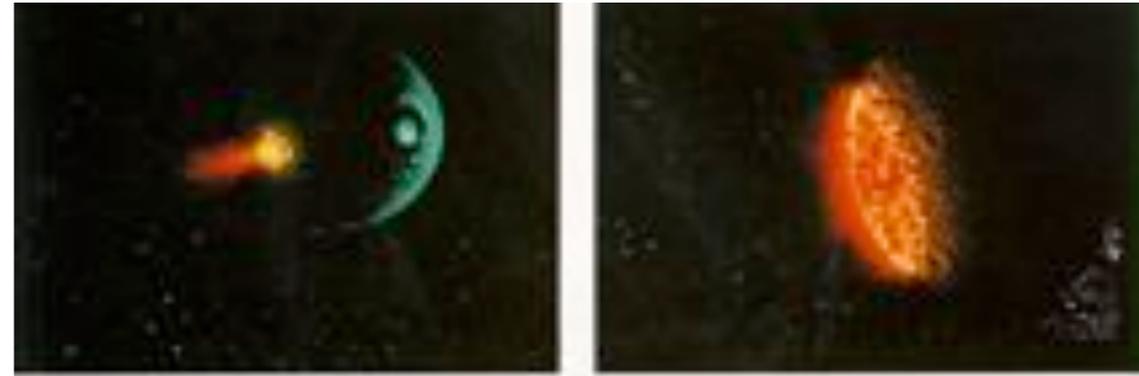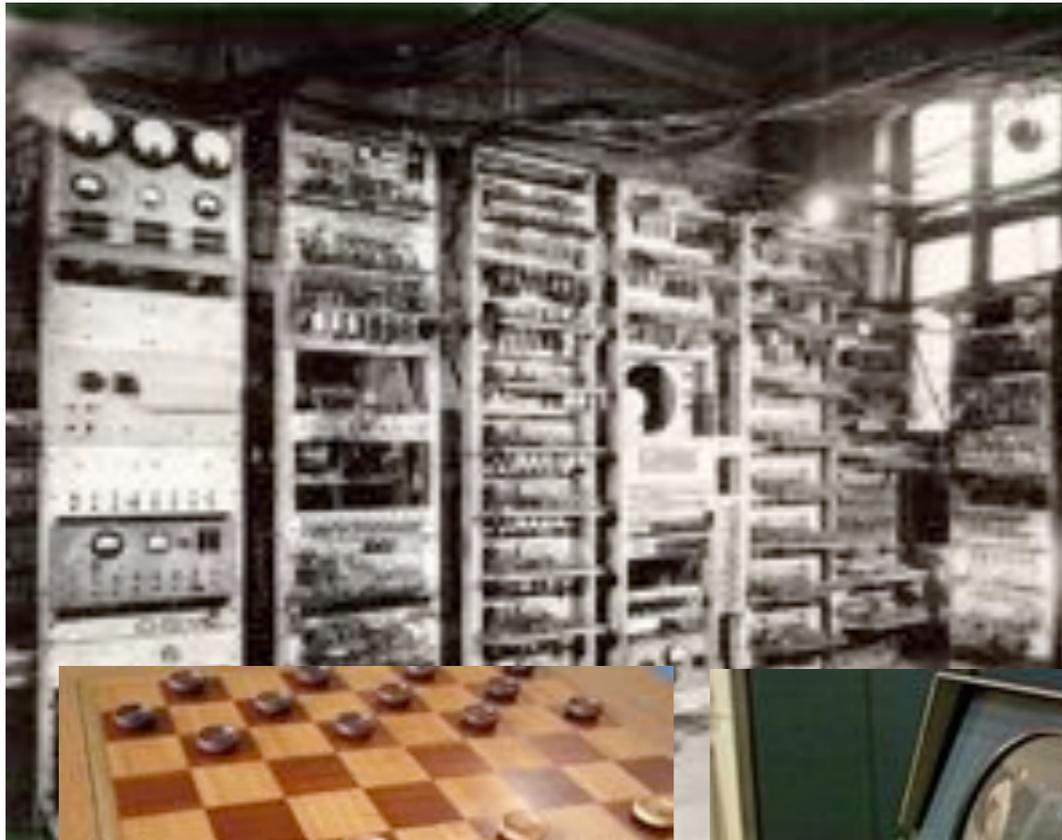# Lessons Learned

Lessons Learned

DUCK DUCK
    YOU ARE MY LITTLE AFFECTION:
MY BEAUTIFUL APPETITE: MY EAGER
HUNGER. MY COVETOUS LOVE LUSTS
FOR YOUR INFATUATION. MY YEARNING
ANXIOUSLY CLINGS TO YOUR FELLOW
FEELING.

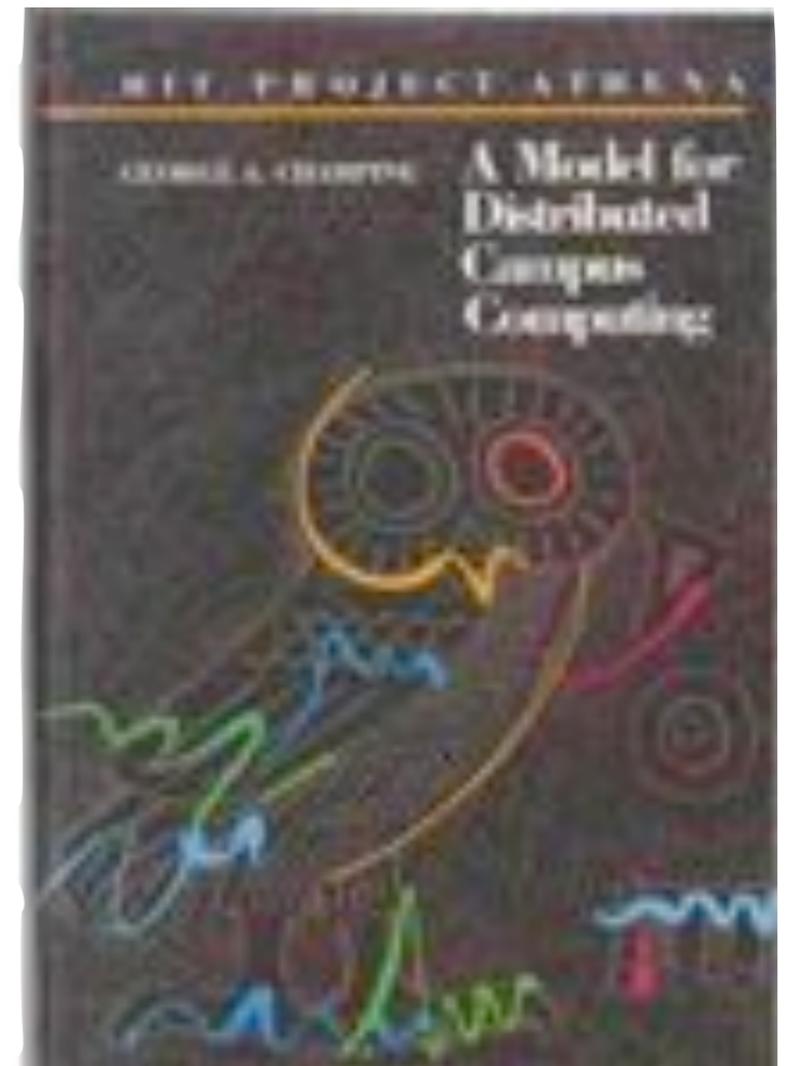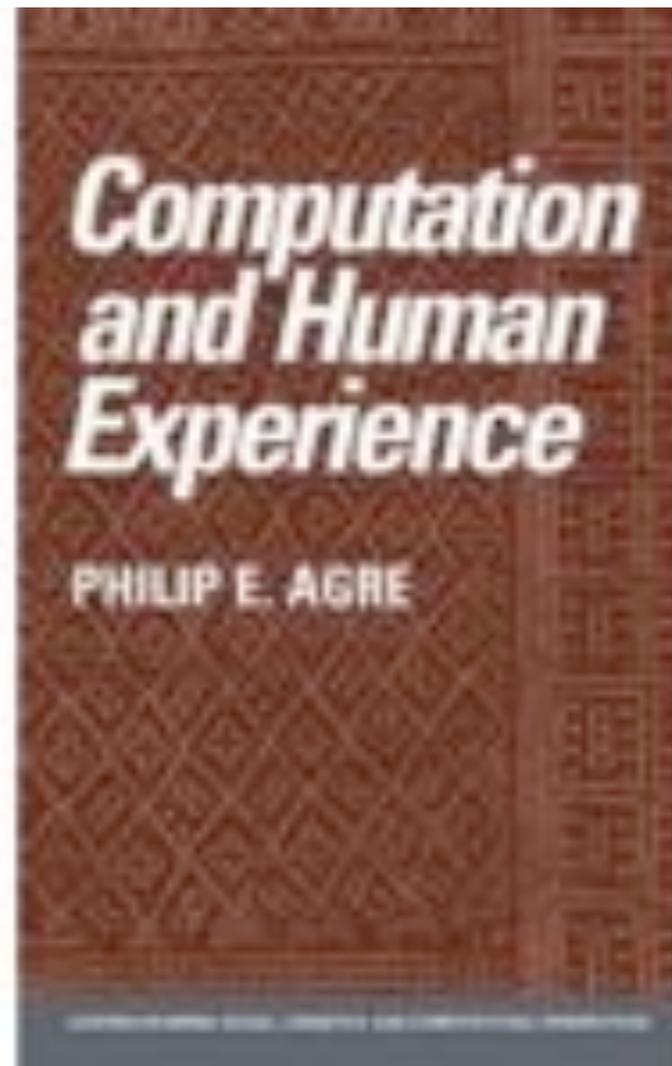   YOURS EAGERLY
                        M. U. C.

We've had digital art since we've had digital computers

Lessons Learned

We've had digital interactive entertainment
since we've had digital computers, and
made media with computer systems for decades

Lessons Learned

We have well-developed examples of humanities interpreting and contributing to the design of computational systems for decades

Lessons Learned

- What lessons have we learned?

- What recommendations might we make, based on these lessons?

- What future projects should we do, based on these lessons, or to communicate these lessons?

Theme 2: Operationalization and Beyond

Or, how to get arts/
humanities ideas into
computational systems
— and use systems to
think about arts/
humanities ideas

Operationalization and Beyond

Score : Overall Composition
Balance + Thirds + Spacing
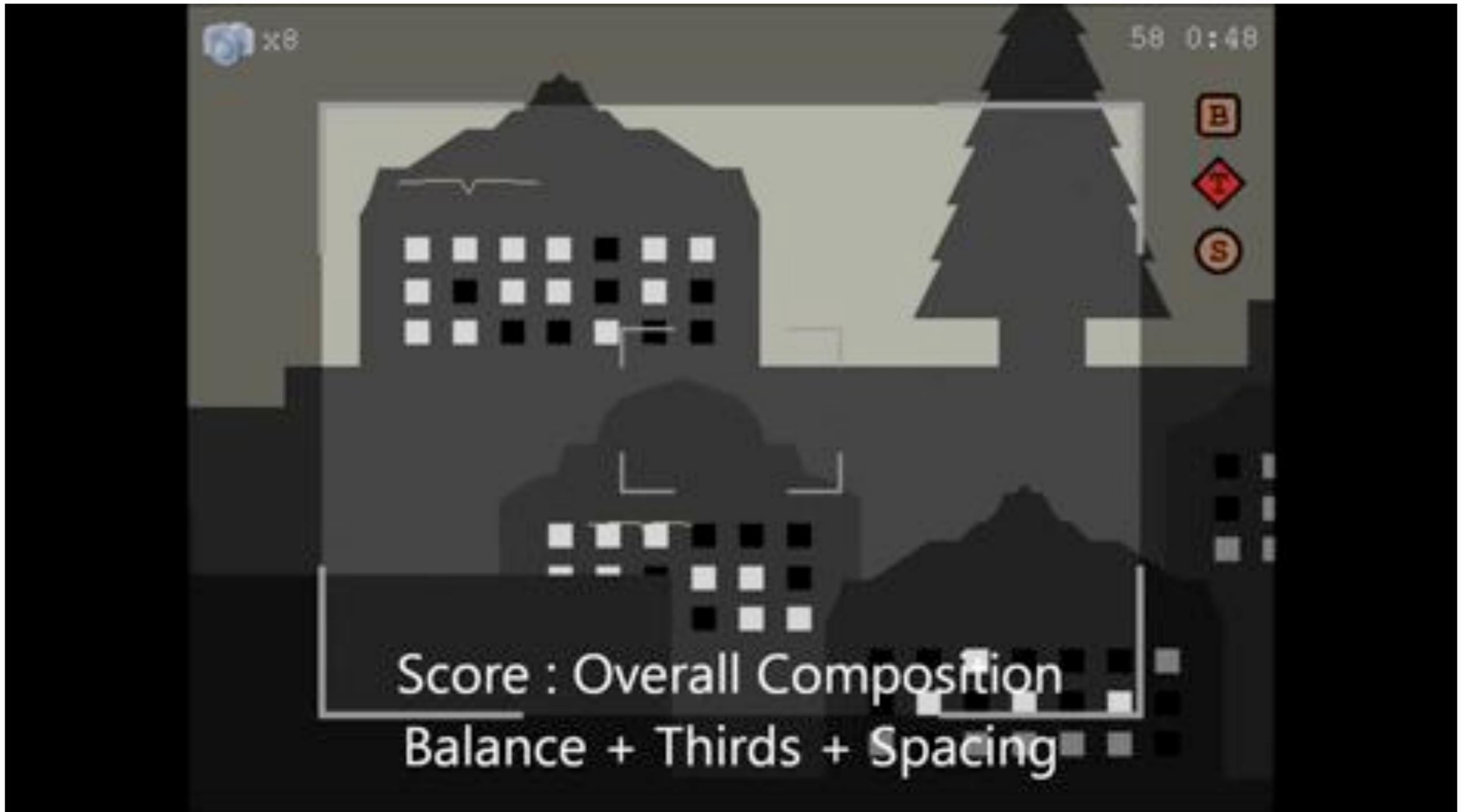
Operationalization and Beyond

Models from other arts point to powerful possibilities for computational systems — building and interpreting those systems may change how we see their inspirations

Operationalization and Beyond

- What are different ways that computational systems can embody/express arts and humanities ideas?

- How can building computational systems give us new ways to reflect on arts/humanities ideas?

- What are powerful approaches for interpreting the ideas computational systems express?

Theme 3:
Guiding and Evaluating

Computer science has tools for evaluating things like efficiency.

How do we guide and evaluate work as meaningful media?

Guiding and Evaluating
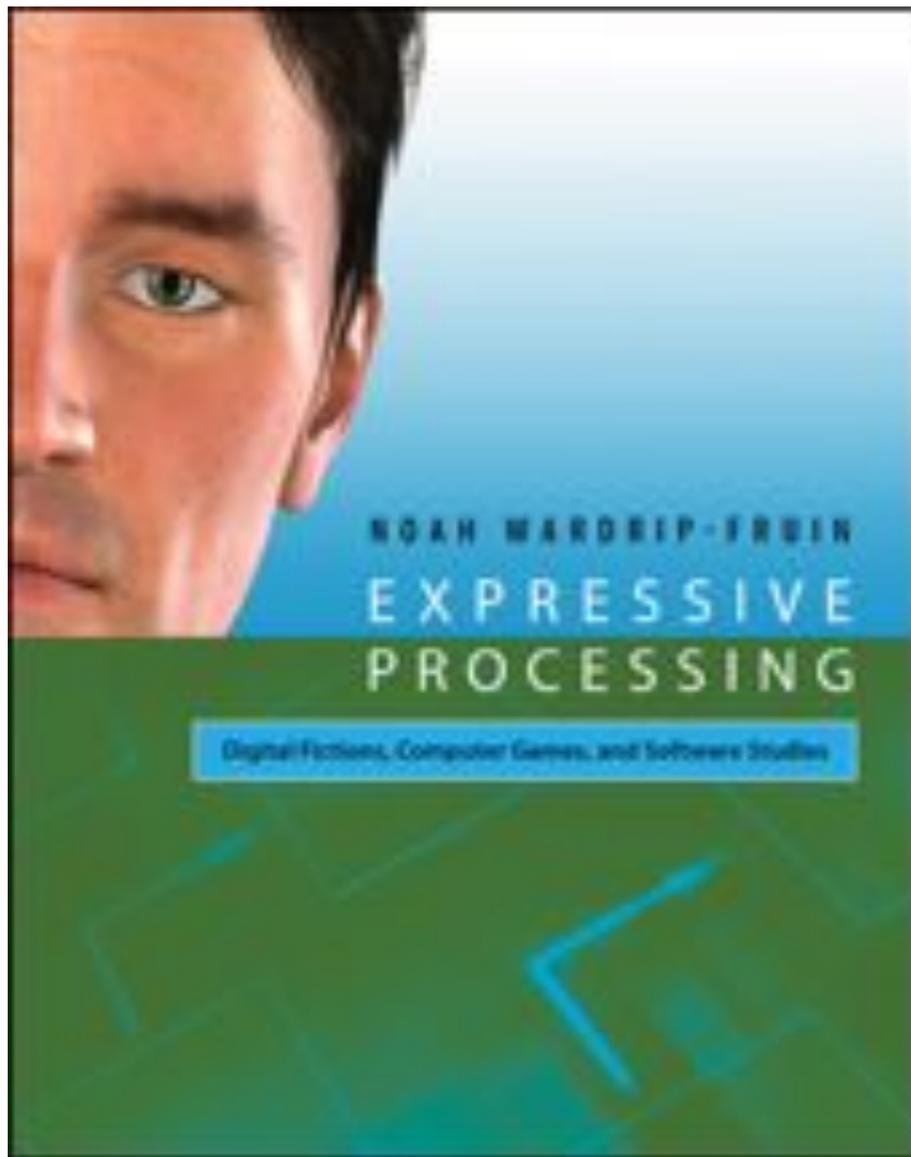
Drawing on software studies, I've tried to use *close reading*

Guiding and Evaluating

- What methods are we using (officially and unofficially) for guiding and evaluating?

- What new ones should we be trying?

- How can we evaluate our new strategies? Do they help make more compelling work?

- How can we change the field to make successful strategies more widespread?

Theme 4: Media Technology Innovation

The arts and humanities suggest things current media technology can't reach.

What are strategies for inventing this next-generation media?

Media Technology Innovation
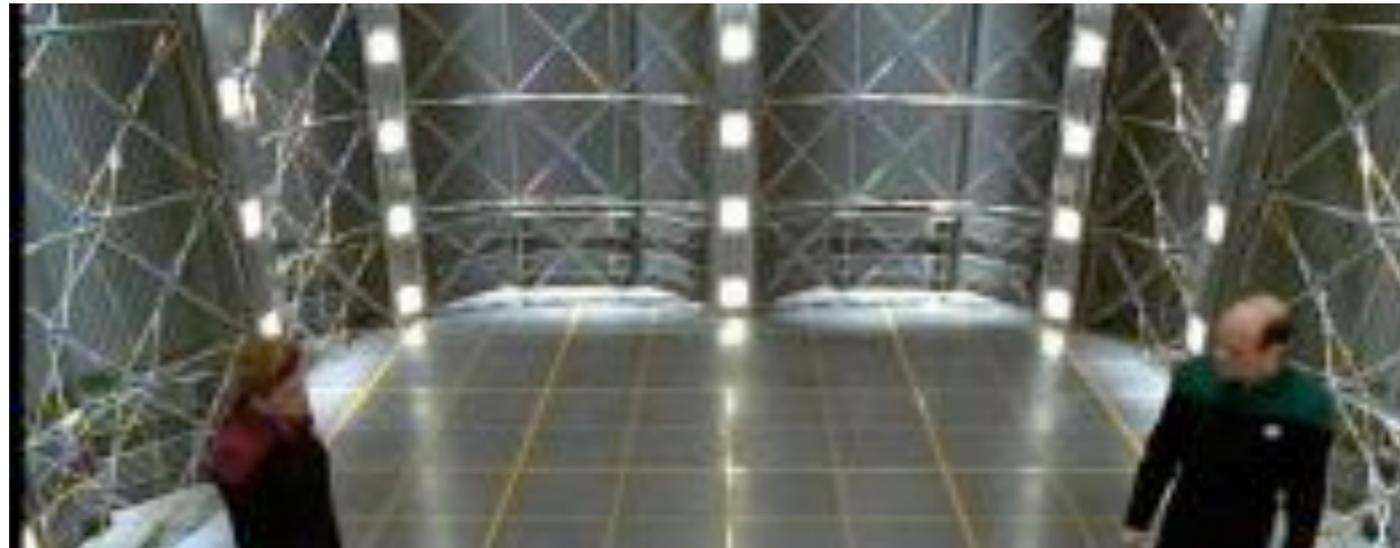
Media Technology Innovation

- What role do arts/humanities inspirations play in different technology development approaches and contexts?

- How can we form and support teams to accomplish this work?

- How can media technology move forward — incrementally and radically?

Theme 5:
Field Building Models

How can we build projects that will transform the field?

Especially when things are built for the projects we have now?

Presentations of three projects that are models of potential transformation

How we will use time

How we will use time

- Framing session for each theme — presenting, interrupting, discussing

- Breakout groups for first four themes:

  - Recommendations to specific constituencies for moving this work forward, related to current theme

  - Potentially transformative projects, related to theme, bigger than what one lab or institution might do on their own

How we will use time

- Sharing 2-3 developed project ideas after each breakout group

- Mixing together in different groups: on shuttles, at receptions, during meals — this is time to talk about potential collaboration

- Discussion with field builders

- Full group project focusing

- Small group project development

What kinds of projects?

Projects I couldn't imagine on my own!

But here are some examples anyway

Next-generation transmedia narrative

- Retain mystery, spectacle, real-world integration & social interaction of ARGs

- Add meaningful choices and individualized narrative progression — CS research toward new arts tools based on humanities interpretation of genre structures

- Dynamically matchmake each person with others who should be part of their story?

Interdisciplinary computational thinking

- Retain tools with powerful encapsulation of computational thinking approaches

- Add focus on expressing student ideas about the world, requires art/design critique and new tool possibilities

- Add historical context and critical interpretation of what processes express — student ones and those found in society

Software scholar's workbench

- Retain careful interpretation of computational systems and platforms

- Add the ability to snapshot, share, and cite system states. Tools for extracting system resources, exploring source repositories, citing files/lines, decompilation, visualizing activity, tracking shared dependencies, etc

- Prototype "self-interpreting" critical editions of operating software?

Now for theme I