

How to Design

FIT2099: SEMESTER 2 2018

Where were we?

- Discussed when to design
- Discussed what we should produce when we design
- Discussed some ideas on how to evaluate a design
- But what about ...how?

Page-Jones on “how to design”



African field cricket

Source: <https://commons.wikimedia.org/w/index.php?cu>

What's the procedure for making a design?

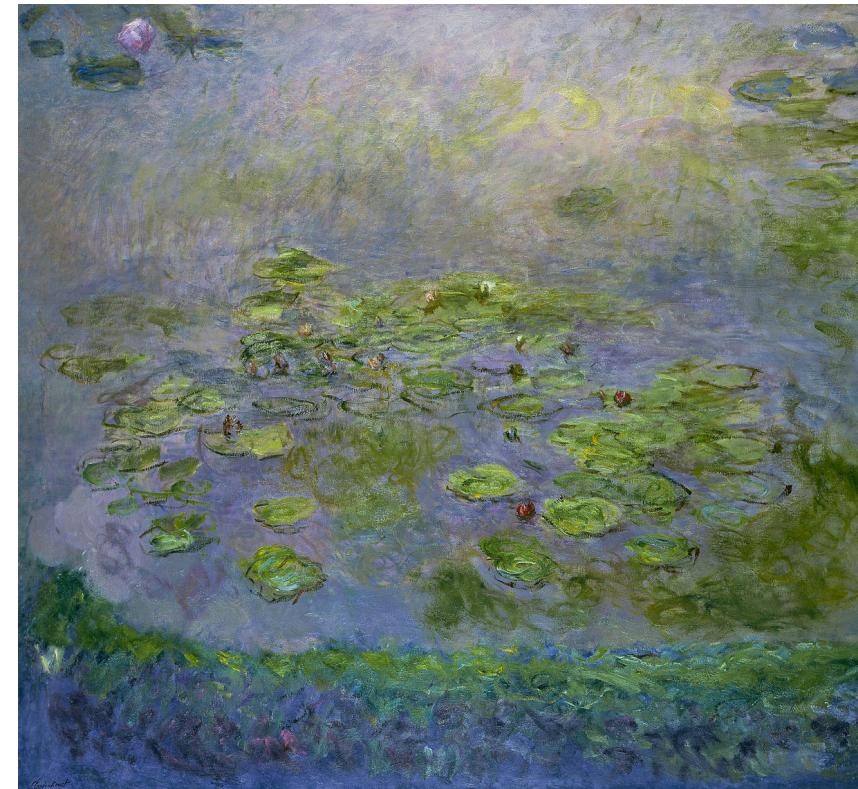
- Let's answer by asking another question:

How do you make a painting?



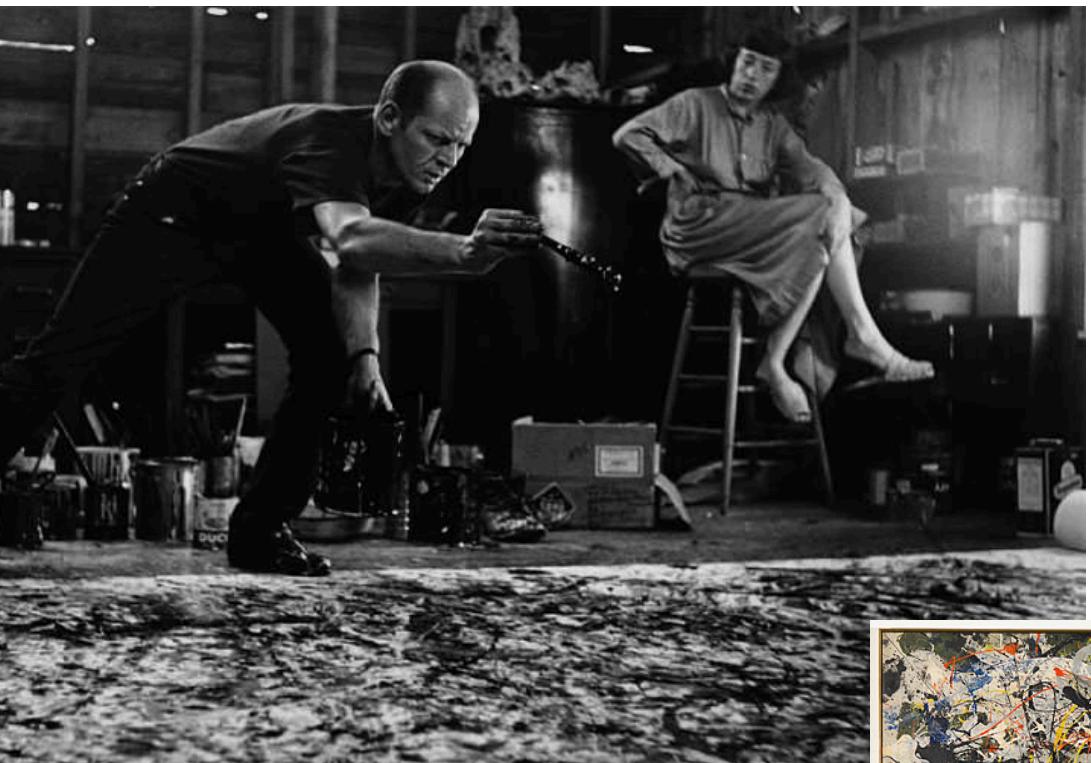
John Singer Sargent, *Claude Monet Painting by the Edge of a Wood*, 1885.

Source: <http://www.tate.org.uk/art/artworks/sargent-claude-monet-painting-by-the-edge-of-a-wood-n04103>

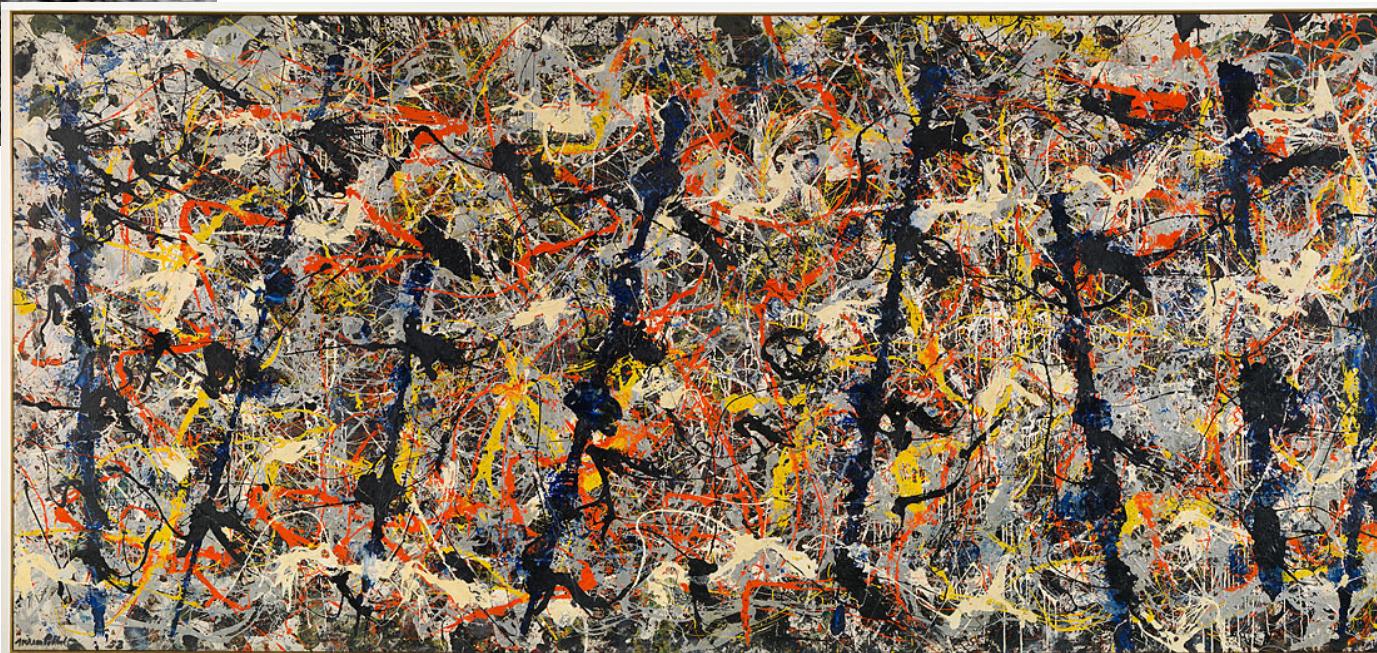


Claude Monet, *Nymphéas (Waterlilies)*, 1914-1917

Source: <http://artsearch.nga.gov.au/Detail.cfm?IRN=64894>



Jackson Pollock at work.
Source: www.jackson-pollock.org



Blue Poles, Jackson Pollock, 1952

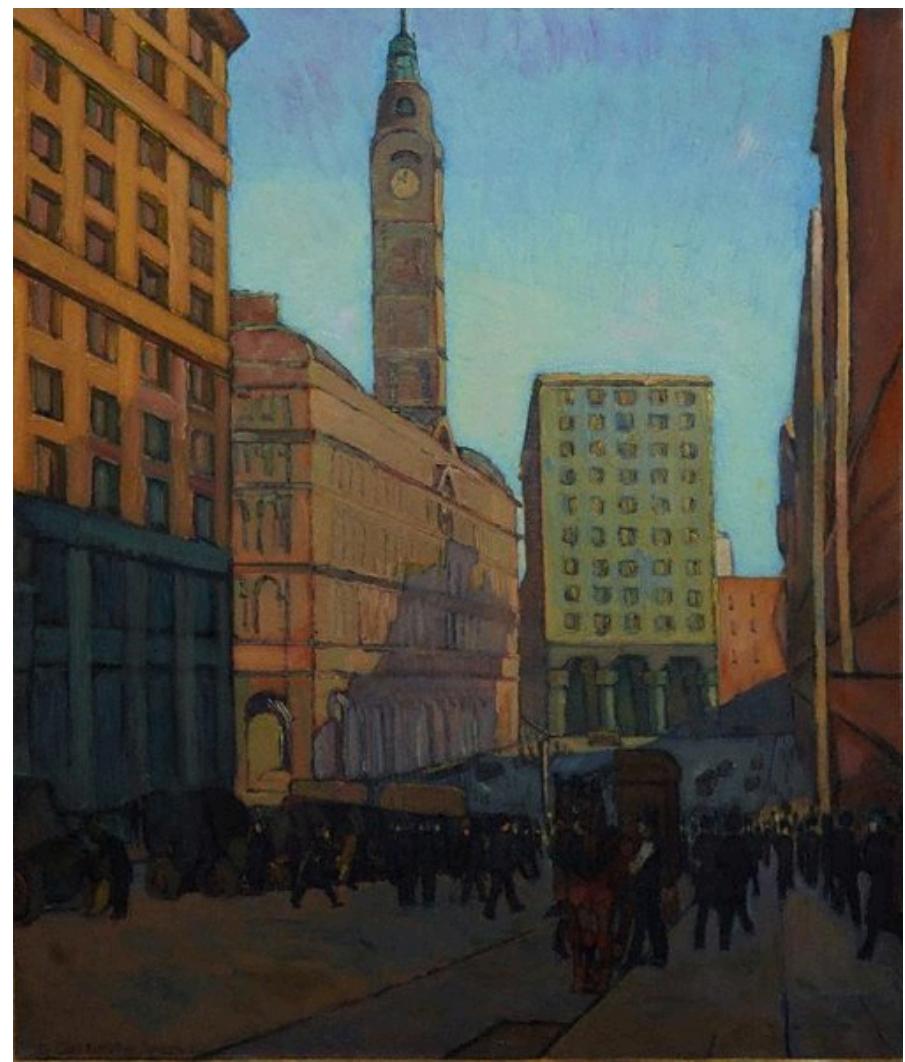
Source: <http://artsearch.nga.gov.au/detail.cfm?irn=36334>



Grace Cossington Smith,
sketched (working drawing for 'Centre of a City')
<http://cs.nga.gov.au/Detail.cfm?IRN=133535>

Grace Cossington Smith *Centre of a city (c.1925)*

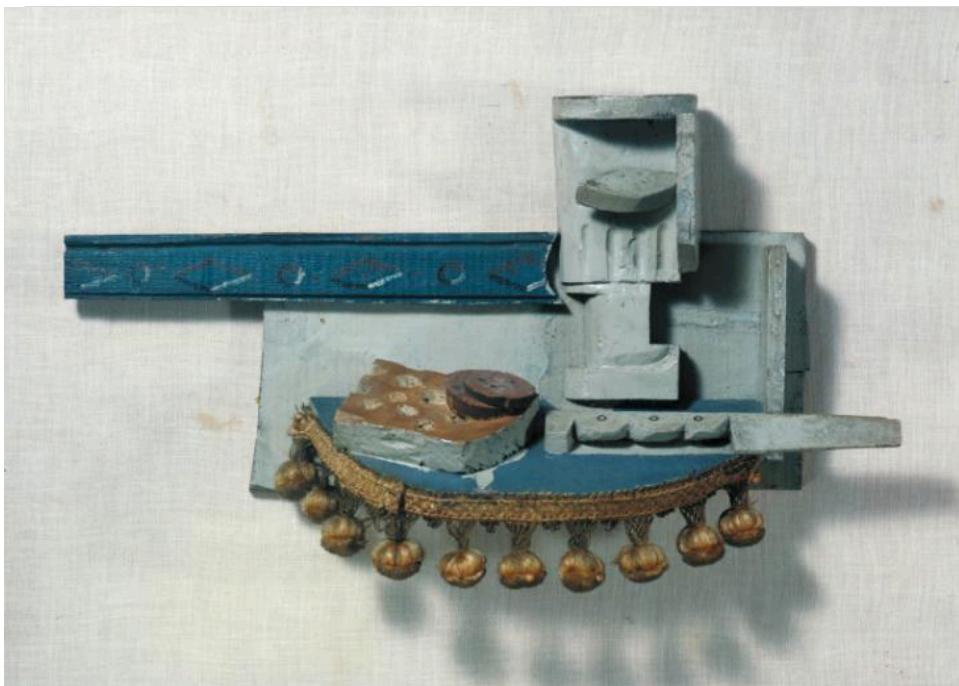
Source: <https://www.artgallery.nsw.gov.au/collection/works/299.2002/>



Three artists...

- Three completely different techniques
- All very successful painters

Working in different media



Pablo Picasso, Still Life, 1914, (mixed media)

Source: <http://www.tate.org.uk/art/artworks/picasso-still-life-t01136>

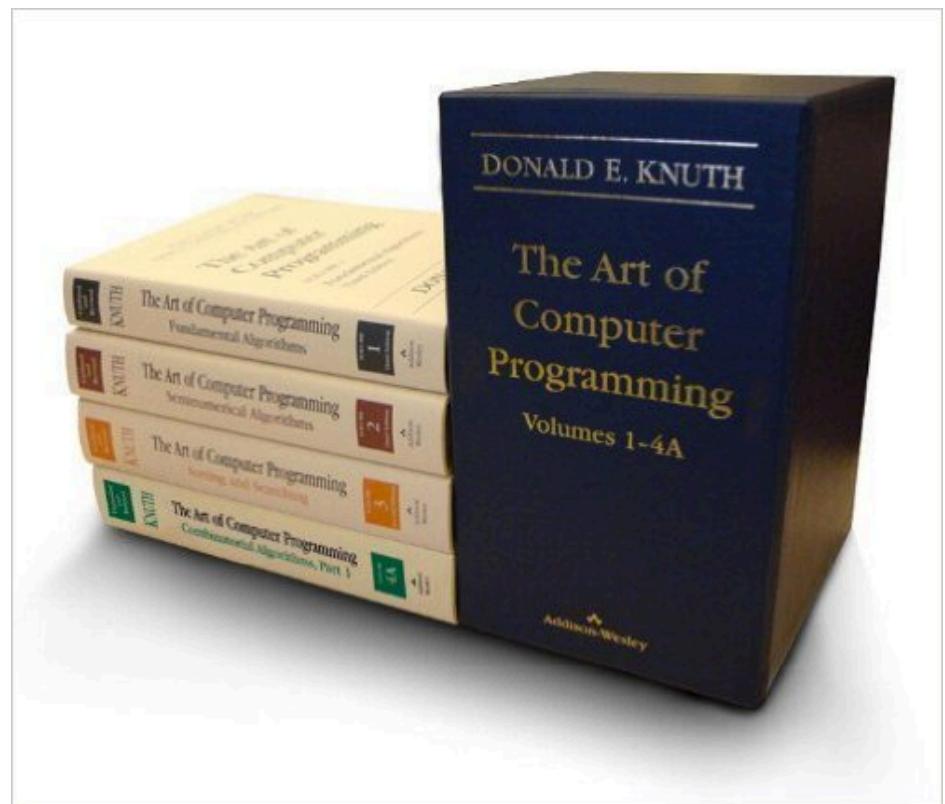


August Rodin, *The Thinker*, 1882 (Bronze)

Source: <https://www.ngv.vic.gov.au/explore/collection/work/3582/#>

So... software?

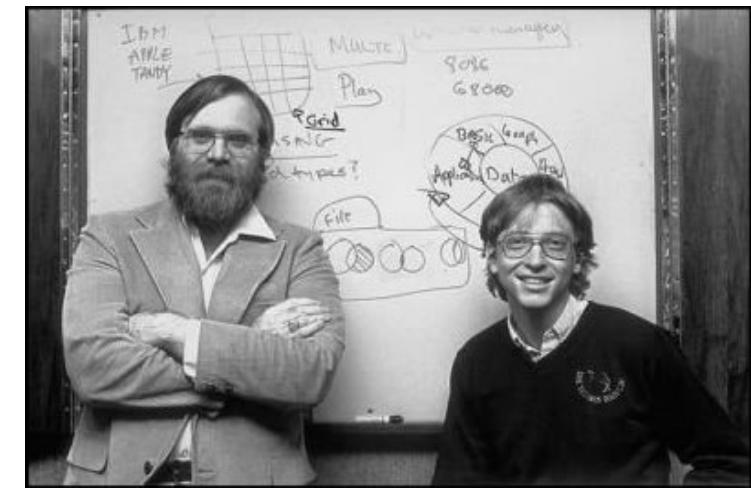
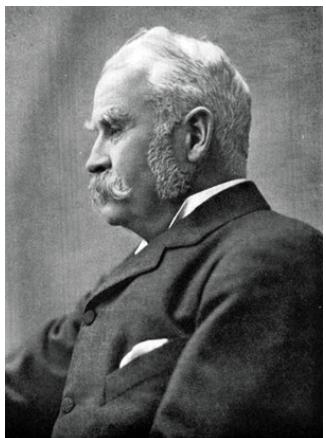
- Making software designs is a creative activity
- We design many different sorts of software
- Designers use a variety of *explicit* techniques to assist
- Like any creative activity, it requires skills
 - Which you get through practice



Where do you start?

- Start by *understanding the problem domain*
- Draw models of the problem domain, e.g.
 - Conceptual or Domain Class Diagrams
 - Activity Diagrams
- Can be evolved

Collaborative design



Brainstorming

- General approach to solving problems requiring creativity in groups
 - Popularized by Alex Faickney Osborn - advertising executive and author
- Key “rules of brainstorming” (as originally specified):
 - Go for quantity
 - Withhold criticism
 - Welcome wild ideas
 - Combine and improve ideas
- Can be a good way to start
 - And you can throw out the chaff later
- “Model storming” is a software-specific variation – see reading
 - Perhaps not so much withholding of criticism

Start at the top

- Top-down design:
 - Start with high-level problem
 - Divide into sub-problems
 - Perhaps recursively
 - Design to solve those
 - Put it together...
- A very common approach in many branches of engineering
 - Can lead to repetition due to repeated sub-problems if not careful

Start at the bottom...

- Start with a small problem that you can solve
 - Design a solution to that
 - Do a few more...
 - Start putting them together
 - Voila...a solution!
-
- Can be useful to do a few “spikes” at the bottom level to gain understanding, and then to switch back to something more like top-down design – perhaps multiple times

Scenario-based design

- Have some scenario(s) that the thing being designed needs to support
 - Storyboard, use case, activity diagram, plain text, etc.
 - This may come out of requirements or analysis (depending on whether thing is “the system” or some small part of it)
- Work through your scenario(s)
 - Trace through your design as it stands
- Modify/rework design to support scenario effectively
 - Keep quality properties in mind
- Repeat with additional scenarios

CRC cards

- You don't need a special notation for doing this
- But some people find an alternative notation useful at some points
- Class-Responsibility-Collaboration cards
- Invented by Ward Cunningham as an OO design teaching tool

CRC cards

Class (name)	
Responsibilities	Collaborators

A work-in-progress CRC card

Unit	
Track enrolment	Student
Report outcomes for a student	University

Using CRC cards

- We start with only one or two obvious cards and start playing “what-if” (with scenarios)
- If the situation calls for a new responsibility, either
 - add the responsibility to one of the objects, or
 - create a new object
- Add collaborations as we go
- If design can be improved, rewrite the card(s)
- Use a magnet to stick them on a whiteboard, if available

CRC cards: working through scenarios

- Have different people “play the object” during a scenario
- Messages between objects -> “Hey Unit, gimme a list of students enrolled in you...”
- pick up the card whose role they are assuming while “executing” a scenario
- When a new responsibility emerges, add it!

More on using CRC cards

- If card becomes too full:
 - copy the information on its card to a new card
 - Express responsibilities more succinctly/abstractly
- If rewrite not possible:
 - Split object up

CRC Cards: top-down or bottom up?

- Whatever works for the group!

Design with the cards tends to progress from knowns to unknowns, as opposed to top-down or bottom up. We have observed two teams arriving at essentially the same design through nearly opposite sequences, one starting with device drivers, the other with high-level models. – Kent Beck and Ward Cunningham, , *A Laboratory For Teaching Object-Oriented Thinking*

Are CRC cards enough?

- If you're Kent Beck, probably...

Certainly XP de-emphasizes diagrams to a great extent. Although the official position is along the lines of "use them if they are useful", there is a strong subtext of "real XPers don't do diagrams". This is reinforced by the fact that people like Kent aren't at all comfortable with diagrams, indeed I've never seen Kent voluntarily draw a software diagram in any fixed notation –Robert C. Martin, *Is Design Dead*

- But...not a good way to communicate to “outsiders”:

We know of one case where finished cards were delivered to a client as (partial) design documentation. Although the team that produced the cards was quite happy with the design, the recipient was unable to make sense of the cards out of context. – Kent Beck and Ward Cunningham, *A Laboratory For Teaching Object-Oriented Thinking*

Connascence and encapsulation?

- CRC card process helps with encapsulation
- CRC cards encourage small objects with clear responsibilities
- Doesn't *guarantee* a good design
- *Always* keep design principles in mind

Summary

- Software design is a complex creative activity
- “How” to do it is often poorly articulated
- Designing in small teams helps
 - brainstorming, model storming, CRC cards, UML on whiteboards
- Top-down, bottom-up and working through scenarios
- CRC cards – tool to help work through scenarios to evolve a design
- Need lots of practice!