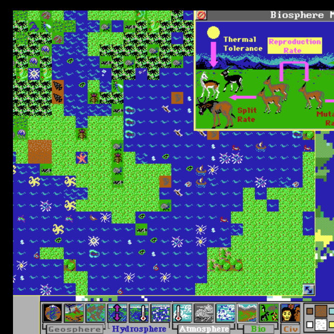
Spore's Magic Crayons: Playthings that Augment the Human Mind

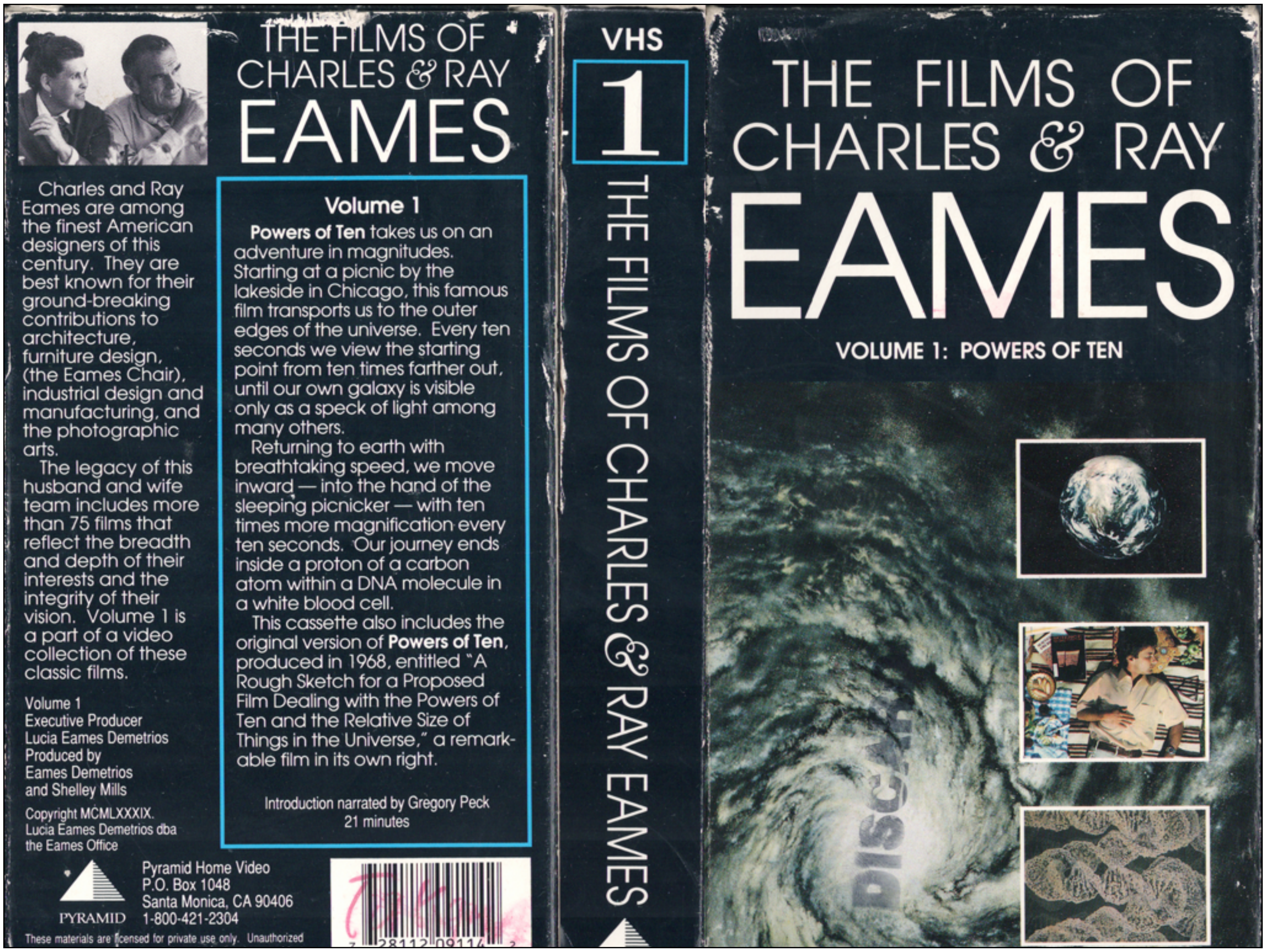
Chaim Gingold • UC Santa Cruz

1,805
Name Your Creature

Will Wright


GAMESPOT


Thermal
Tolerance
Reproduction
Rate
Split
Rate
Geosphere
Hydrosphere
Atmosphere
Bio
Civ

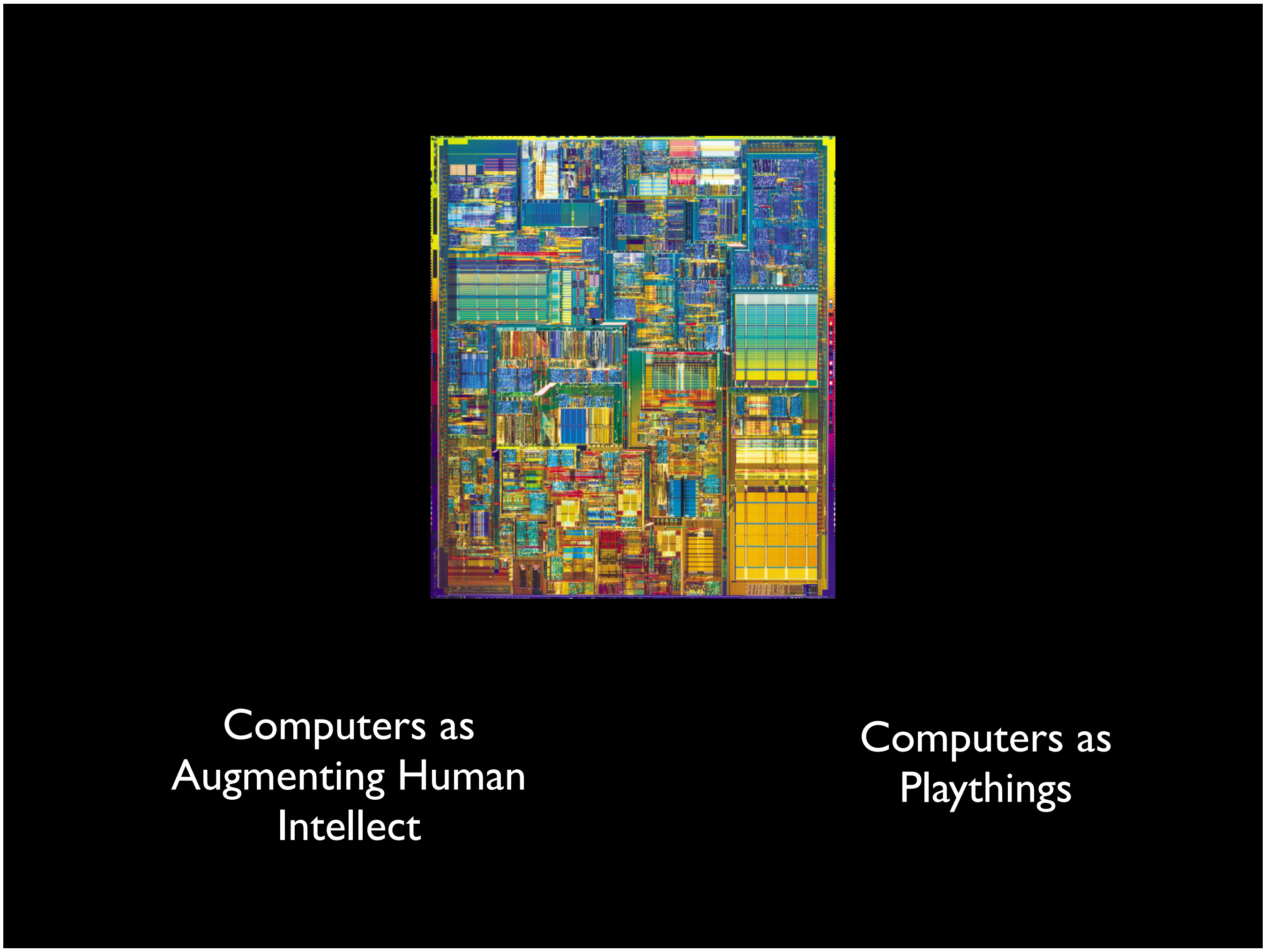


Will has an affinity for highly creative, ambitious, projects. Following The Sims, Will decides to do a game inspired by the Eames' Powers of Ten video, Drake's equation for the probability of life occurring in the universe, the life of the cosmos, Carl Sagan—a host of interesting ideas.

Which becomes the game Spore, which is released in 2008. The evolution of Spore itself is an interesting story that isn't our subject. I originally joined the project as an intern in 2002, when it had a staff of about 3 people.

In its final form, players would begin their journey as a frozen bacteria crash landed on a sterile planet, and progress and direct their "evolution" through a series of stages from microbes, onto land, acquire intelligence, become civilized, take over the world, and eventually travel to other stars and plants populated by alien creatures, architecture, and vehicles designed by other players.

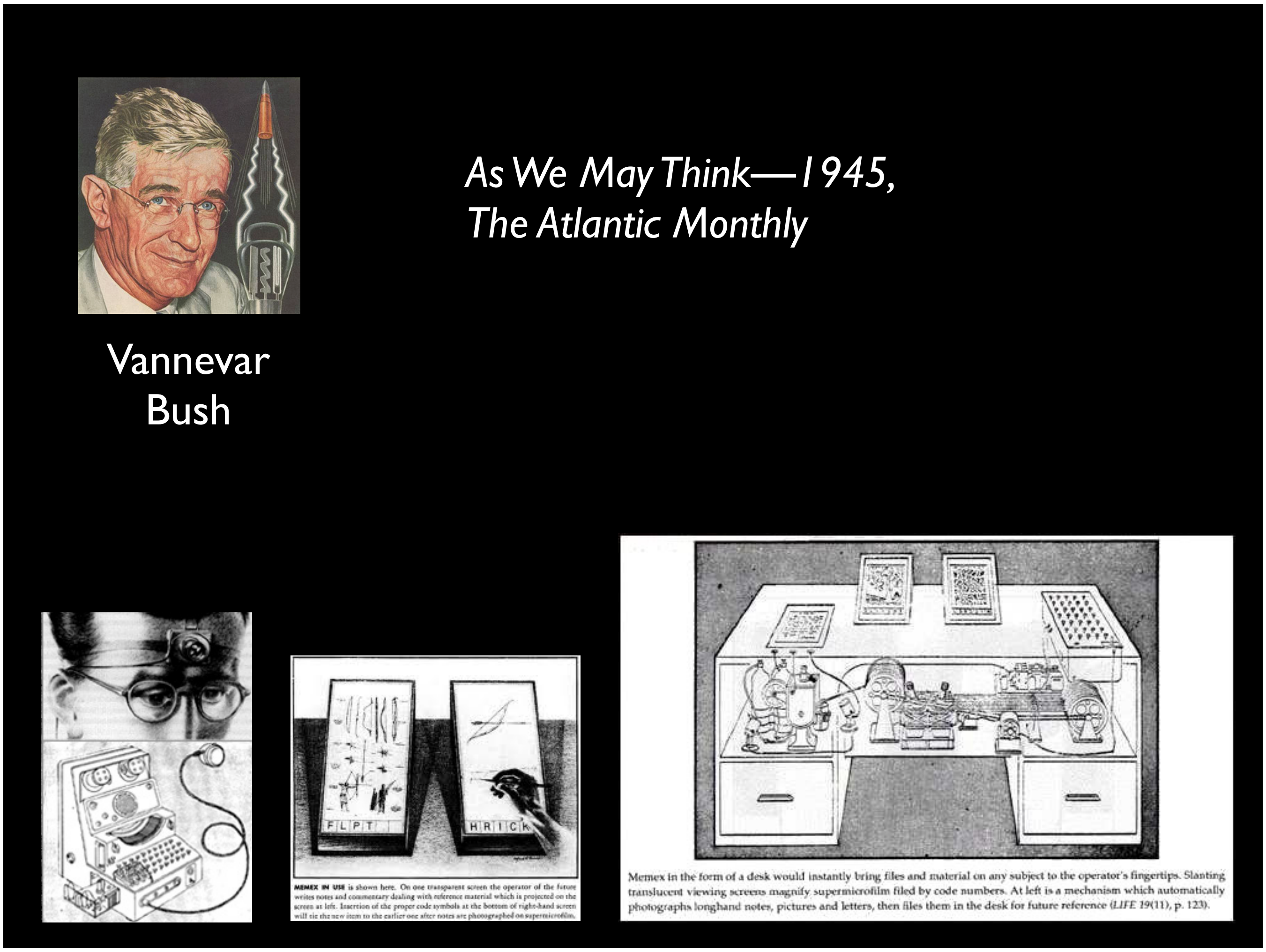
The game had multiple editors in them—tools for making creatures, buildings, & vehicles, which I was the lead designer of.



What I am going to talk about today is the legacy that Spore's editors—and the category of software playful creation tools I call Magic Crayons—are heir to.

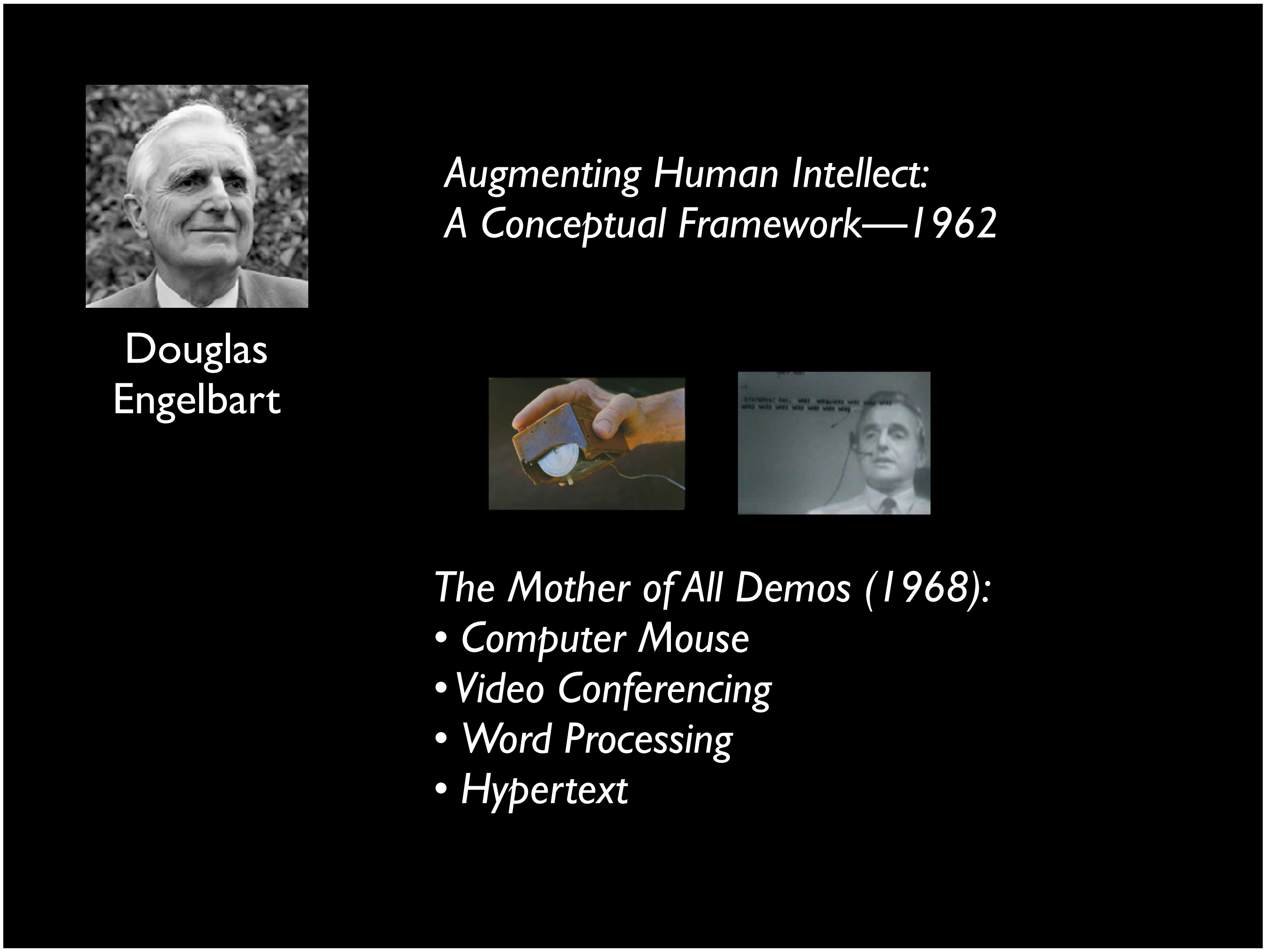
If we look at the history of computation we find that Magic Crayons represent the intersection of two major historical threads, different ideas about what computation is for & how we should use it. One story is of how visionaries saw in the computer—in mechanized computation—the possibility of augmenting our ability to think & solve problems; the other is of computers as toys, devices for showing off the capabilities of these new machines, bringing them to the masses, and entertaining them.

The story I will tell you today is how these strands separate strands developed in parallel, and then intersected, giving us a new class of computer software that I call Magic Crayons.



It's 1945—and the man who directed the US research efforts in WWII, Vannevar Bush, writes an influential article in the atlantic monthly. He argues that scientists must now direct their efforts to peaceful applications of technology. This article is best known for the Bush's depiction of the "memex," <click> a device for rescuing the researchers of 1945 from an information overload owing to specialization. The memex, stemming from the human mind's associative memory, enables scholars to link items stored in its memory, and take notes—prefiguring, in some ways, the internet. (Of course, we still don't organize our own files on our machines via digital association).

But this article had a tremendous impact, as we will see, by extrapolating from current trends with technology (the information explosion, coupled with miniaturization of storage, fast recall, easy capture), painting fantastic scenarios of they might be used, and arguing that machines, if properly designed to accommodate human beings, could be used to elevate human scholarly capability.

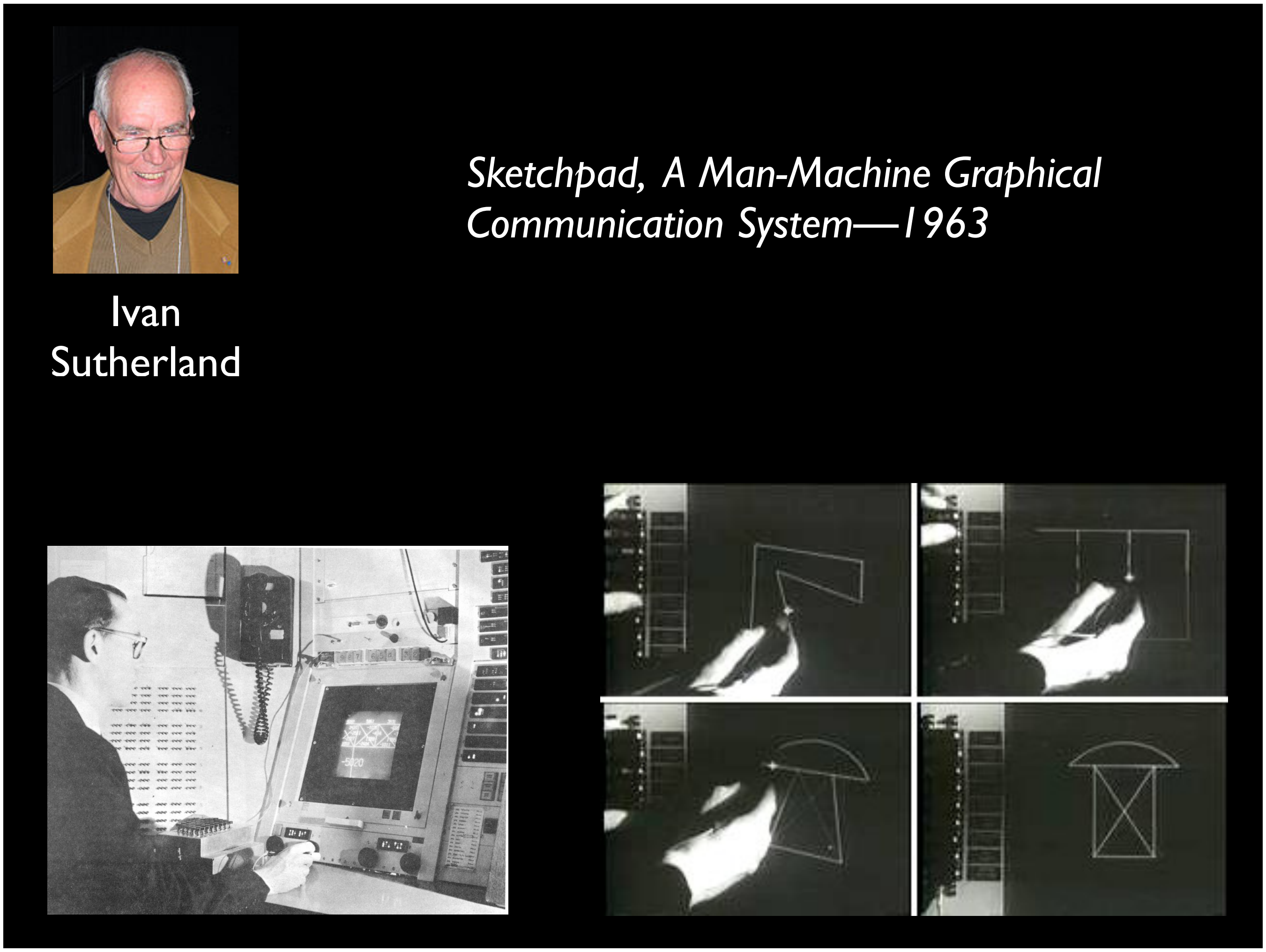


In 1968 Douglas Engelbart, hugely influenced by Bush, writes a research proposal that lands him funding from ARPA, and founds the ARC (Augmentation Research Center) at SRI (Stanford Research Institute). This proposal is a very interesting document which defies summary, but I'll do my best: Engelbart articulates a broad design framework for thinking about how humans, having already been augmented by language, artifacts, and methods, could be augmented farther with computation. Basically, Engelbart argues we can decompose the hierarchy of how humans perform various tasks, all the way from the cognitive to the physical, and experimentally evaluate computational interventions in that hierarchy.

In his proposal, Engelbart struggles with how to depict the possibilities of this new machine to his readers who, like him, are trapped in the computational present. At times he refers to this device with a "clerk" metaphor. This thing, whatever it is, is clearly not an inanimate tool, nor is it an autonomous agent, but something in the middle. Together, the man and the machine, form some kind of cybernetic hybrid.

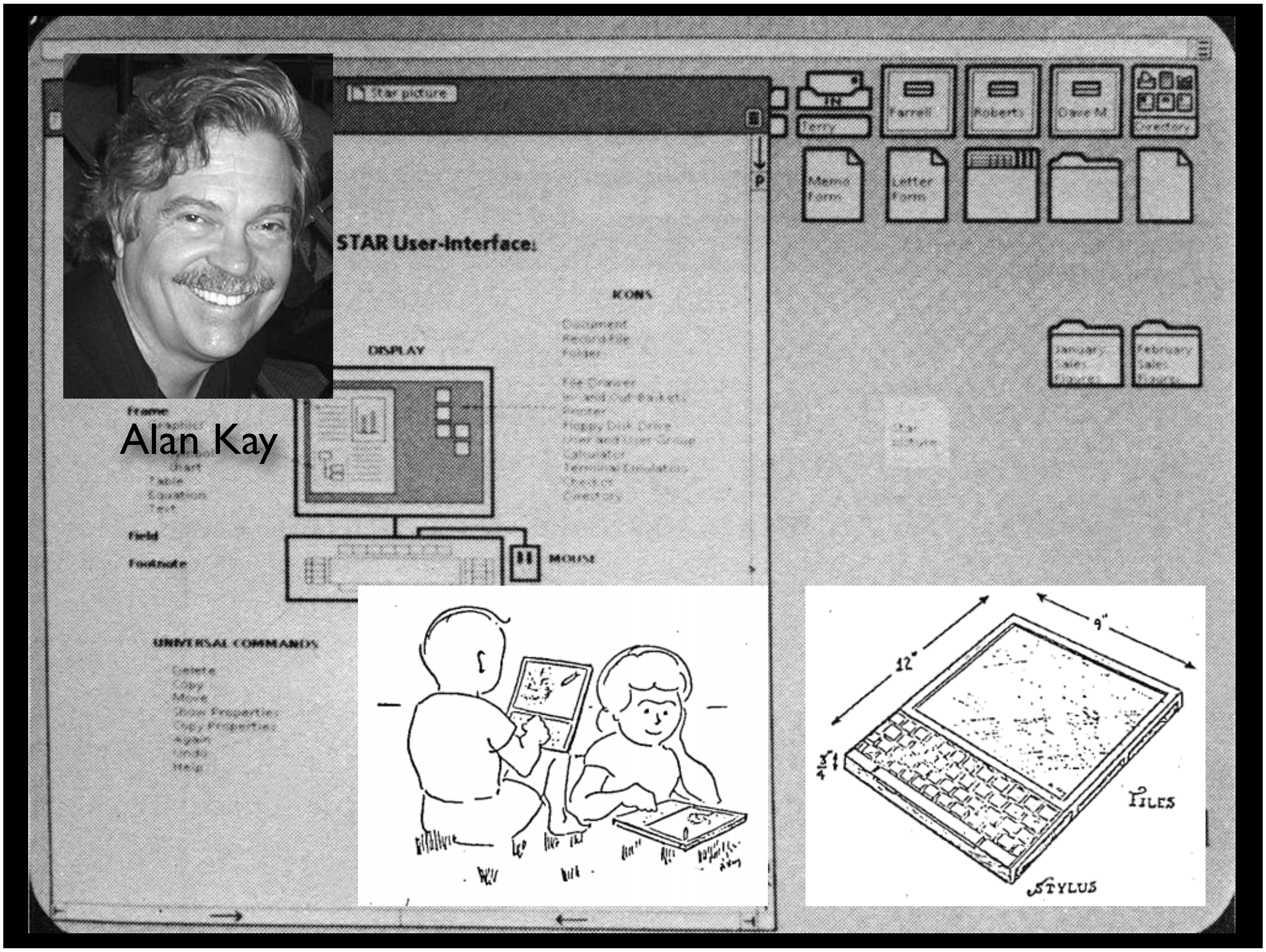
<click>Then, in 1968, in the culmination of this research, he gives the mother of all demos: the On-Line System. Does X, and actually a lot more.

[[clerk: [AI vs IA]. The best answer, I think, is in Sherry Turkle's writings of children assimilating this new computer technology in the late 80's: the machine is neither dead nor alive, but some new category.]]



In 1963, Ivan Sutherland, a PhD student working under Claude Shannon (the father of information theory), publishes his work on one of the world's first graphical user interfaces. The system isn't just a drawing tool. It has a deep structure, a knowledge about the structure of the drawings one builds in it. For example, one can define objects and reuse them—if you update the template, all of the instances change, it can solve constraints—shown in illustration. And it helps you draw perfect circles and other forms.

As a professor at the University of Utah, he has a few notable students: Alan Kay (who we'll talk about in a moment), and Edwin Catmull (co-founder of Pixar).

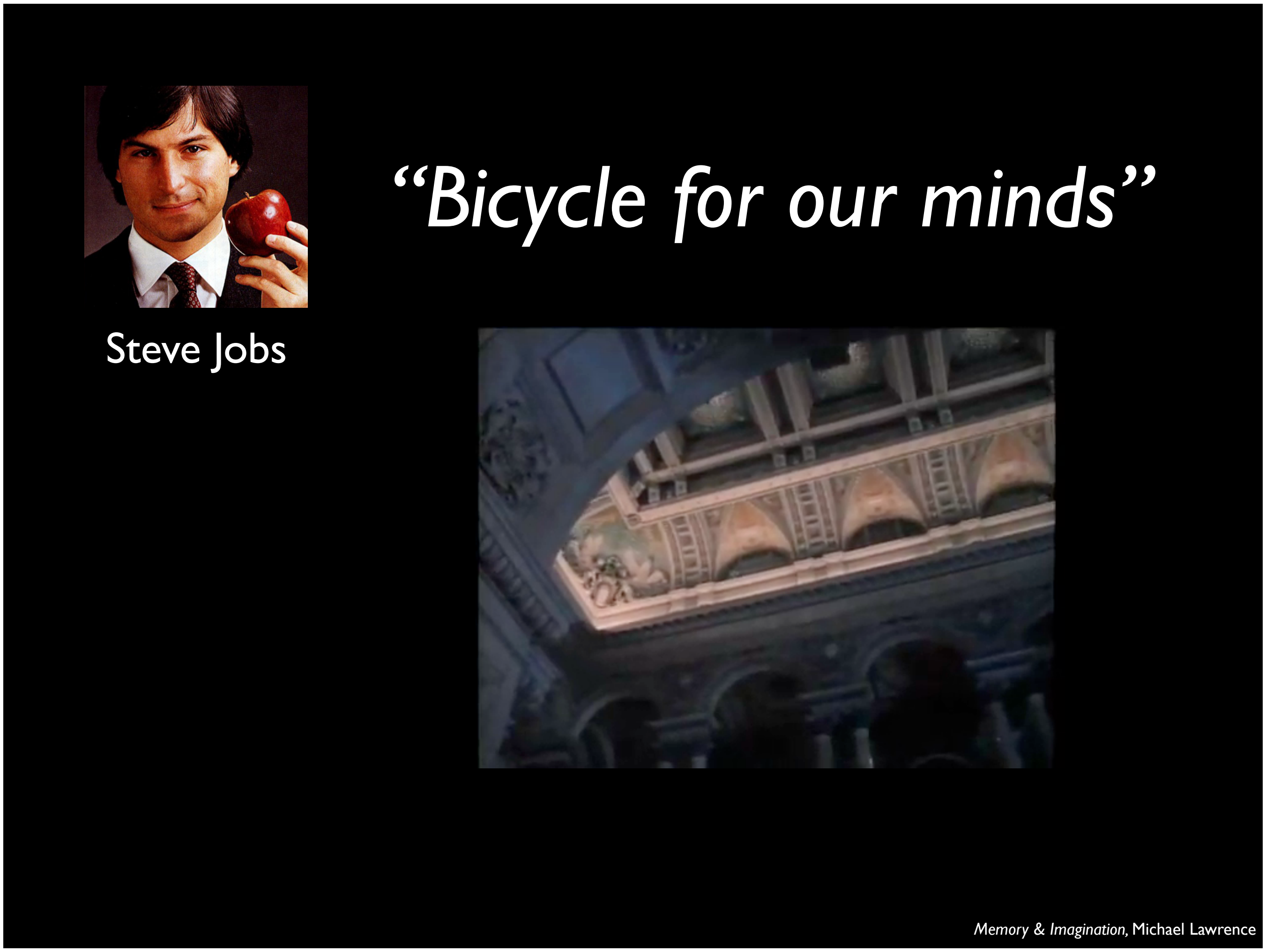


In 1972, Xerox’s PARC research facility starts development of a new computer, inspired by Engelbart’s On-Line System, which prefigures the modern personal computer: it has a graphical user, networking, a mouse, object oriented programming, to just briefly list some key features. This is the design paradigm that Apple then copies and improves upon when building the Macintosh, but it will take decades for many of the ideas developed at PARC to enter the commercial mainstream.

In 1981 Xerox releases the Star, a commercial descendent of this research.

One of the Star’s designers writes in 1981: ”We have learned from Star the importance of formulating the fundamental concepts (the user’s conceptual model) before software is written, rather than tacking on a user interface afterward.”

<click> Also in 1972 Alan Kay, one of the visionary researchers at PARC—and one of Sutherland’s PhD students— writes about the DynaBook—a computer for children that looks an awful lot like an iPad. And actually, in the illustration the kids are playing (and modifying the program for) SpaceWar on their dynabooks—which we’ll come back to.



This technology and design, as we know, is eventually popularized and commodified. The legacy of this work is in the smart phones, tablets, personal computers, and laptops we use today. The thrust of this development, as the master marketer and visionary of modern computation puts it is that the computer is a bicycle for the mind. <click for video>

But bicycles, of course, are more than just instruments for getting from point A -> B. Bicycles are FUN. One can take pleasure trips on bicycles, race them, and play with them in a number of ways. Which brings us to our next story, that of computers as playthings.

This brings us to computer games. The first computer game and video game is somewhat contested. For our story, we will start with Spacewar!—which isn't quite actually the first computer or video game—but occupies a historically valuable turning point that should become clear. Before SpaceWar we have computers playing Nim, Tic-Tac-Toe, a chess playing AI, and even some very rudimentary visual games involving ballistic balls of light.

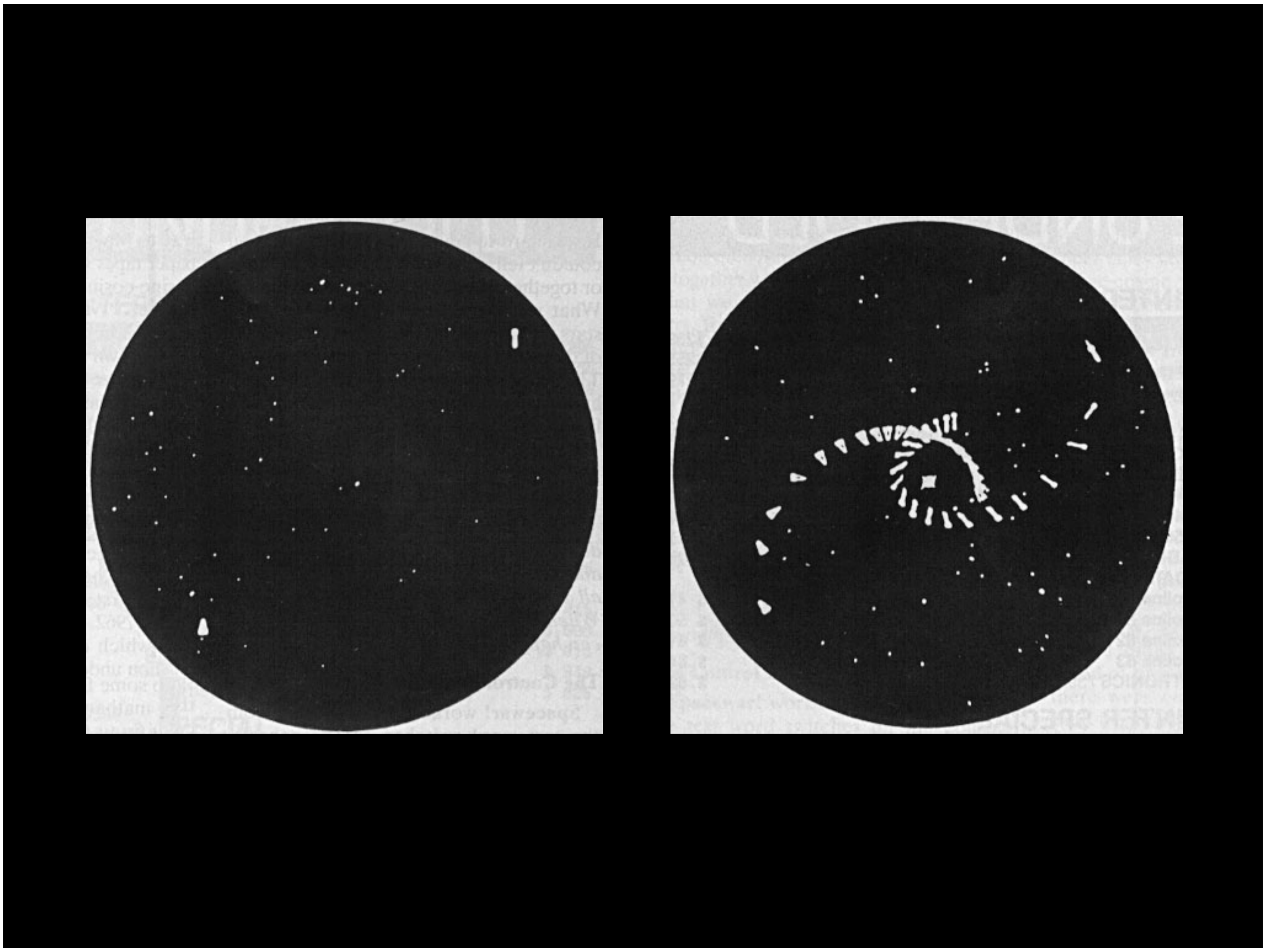
SpaceWar is special because it's akin to the first organism to step on to land. The game is familiar: you would recognize it as a modern video game, and second: it was widely distributed, had a relatively broad following, and a big impact on those who played it; it was an influential game.

[[a) it is played by a (relatively) widespread group of people, b) it foreshadows and inspires a generation of computer game making, c) it is developed under circumstances that most look like the future of computing: a sophisticated (relatively speaking) visual display and interactive computing, and d) we can follow the reasoning of its designers through the process of deciding that the thing they were making was to be a game, and why: a sort of design before and after.]]

This is the IBM 704—a machine that lived in a chapel, and which mere programmers and users did not directly interface with. Computing was reserved for the Gods. Intermediaries were required to interface with The Machine, which was institutionally and technologically not available for individuals to mess around with in real time.

The programmers describe this era of computation in such ritually and religiously charged terms. Which is a nice metaphor.

I see the history of computation in mythical terms: a trickster steals the fire of computation from the Gods, and brings it down to Earth, to help us perform human tasks. Human tasks like listening to music, communication, and playing games.



So when programmers finally got access to real time computers in 1961, and receive this fire—they play around with it and cook up a lot of interesting things. Sutherland's 1963 Sketchpad is an example of this: computers for drawing.

In J. M. Graetz's history of SpaceWar, he describes their new real time TX-0 machine as "a stone's throw from olympus," and it's relative accessibility makes possible a generation of hackers—people who live to play around and experiment with computers to see what they can do. While much interesting research is done on the device, it's not particularly exciting to look at. They find that at open houses people are mesmerized by a bouncing ball demo. Demos proliferate: mice that run through mazes in search of cheese or martinis, and mesmerizing patterns of dots and sound.

When, in the fall of 1961, the PDP-1 arrives—which Graetz calls the "world's first toy computer," and the idea for SpaceWar! is born. What's interesting is that while various playful toys and demos exist, it takes another step before these developers hit upon the idea of making a game. They decide they want to:

1. Demonstrate the machine's abilities to the maximum (which becomes a kind of norm for the game industry).
2. Variety—every run should be different within a consistent framework.
3. And it should involve the onlooker in a pleasurable and active way

—they realize they want to make a game!

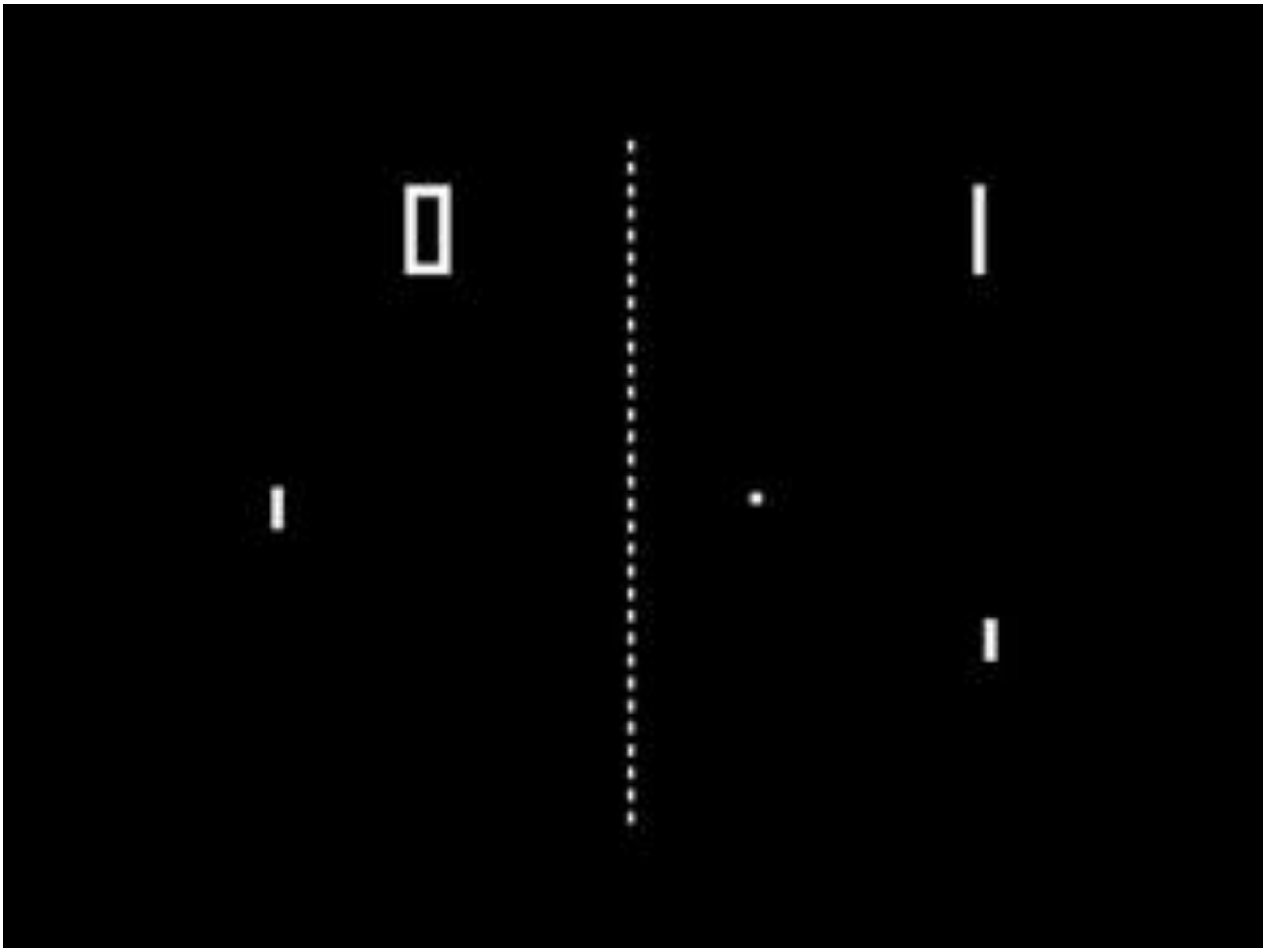
(I want stop and observe that in the gene pool from which computer games emerge we also see things like screensavers the demo scene—other modes of computational play).

Nolan
Bushnell

OK, Nolan Bushnell. As an engineering undergraduate at the Uni. of Utah in the 60's, he played SpaceWar! on the university computers. In his youth, he worked on the midway of amusement parks, hawking and running games of skill and luck—we would seem to have the right materials to produce the man who is regarded as the father of the video game industry.

<click>

In 1971 he begins work on Computer Space—a cheap, consumer, cabinet version of SpaceWar! Understanding the importance of marketing, he puts it in a sleek cabinet, but the game is a commercial flop—the game is too complex.



A few years later Bushnell's company, Atari, releases Pong into the wild, which has much simpler gameplay: "Avoid missing ball for high score."

(Ralph Baer and Magnavox actually design Pong first, but that's another story).

Steve Jobs, incidentally, does a stint at Atari before going on to found Apple. And Wozniak and Jobs—the public co-founders of Apple—do a Breakout game for Atari which Jobs famously stiffs Wozniak on the money for.

Don Valentine, the key venture capitalist behind Apple, is also one of the primary investors in Atari. In due course, a massive industry springs into existence built around the commodification of computation as a means of play. <click> Computers, here, are not devices expanding and extending the mind, but machines for having fun, exploring fantasy worlds, and conquering foes.

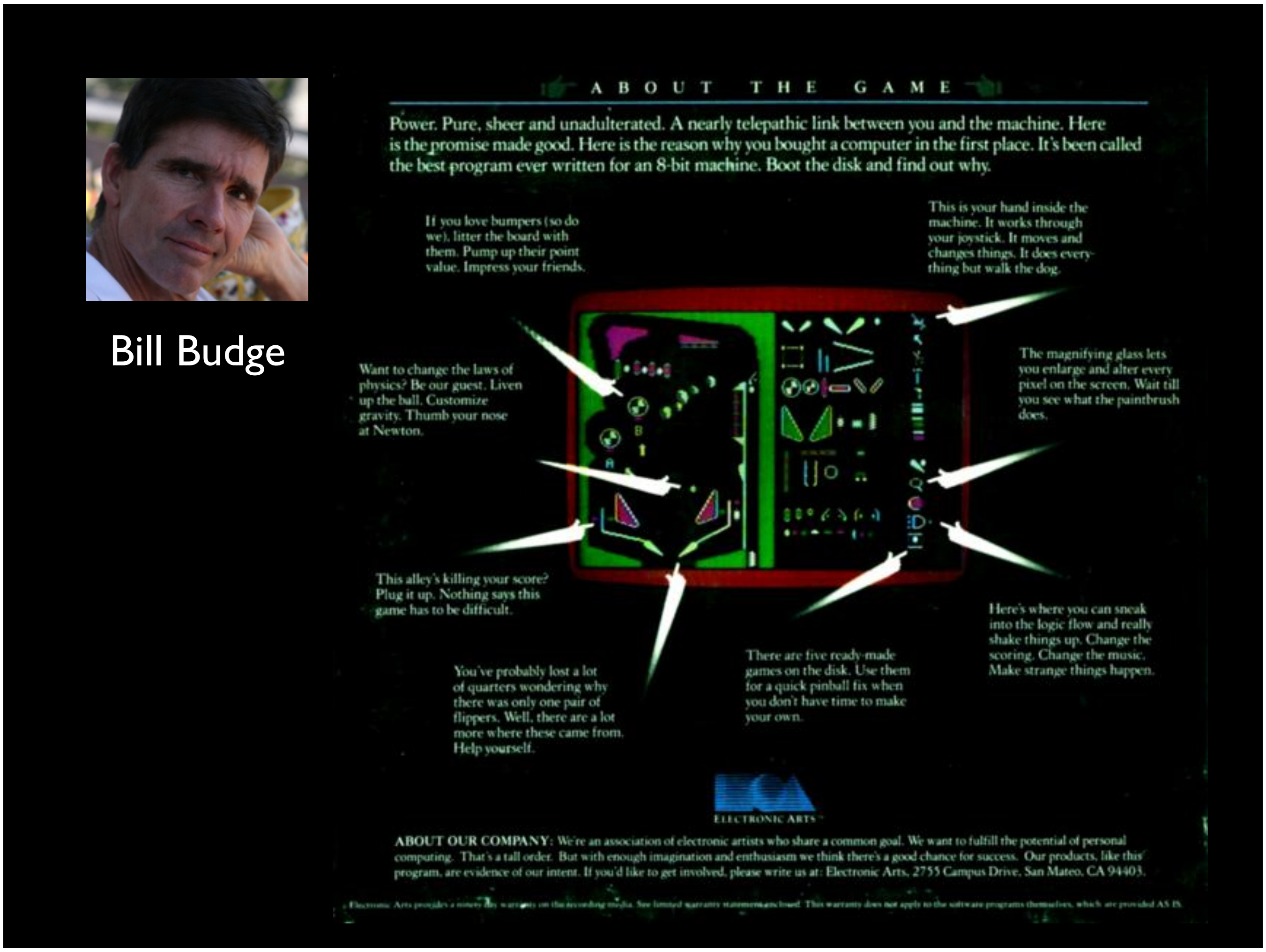
Incidentally, Alan Kay, one of the key researchers at Xerox PARC, goes to Atari in 1980 as a research scientist, where he mentors and makes a big impact on a young game developer named Chris Crawford. Crawford is one of the great reflective practitioners of game design, founds the major conference of computer game development, and popularizes the notion of “nouns” and “verbs” in game design—thinking of the system under design as an object oriented, linguistic construct; an idea we can trace this idea back to Alan Kay and Xerox PARC.

1982, Trip Hawkins founds Electronic Arts—he leaves Apple Computer, and with the backing of Don Valentine of Sequoia Capital (also one of the original backers of Apple), founds Electronic Arts. Hawkins is Apple's 68th employee, and is infected with the idea that computers are destined to be a consumer commodity.

Unlike the EA and mainstream game industry of today, Hawkins envisioned promoting individual artists, like music artists—as the name "Electronic Arts", and this photograph of their first stable of designer/developers, implies. The EA from this era famously asks: can a computer make you cry? Hawkins leads the way to the modern era of computer game packaging & promotion—in well designed boxes instead of plastic bags—that would eventually bury the auteur model.

Incidentally, David Maynard, at bottom center, worked at SRI with Engelbart on The Mother of All Demos. AND he also worked at Xerox PARC on the STAR—he told me that was a lot of traffic between SRI and Xerox. I worked with his son, Jordan Maynard, on Spore, who told me that the games of everyone in this picture were massive commercial successes... except for his dad's. And Will Wright could have been in this picture, as EA was courting him for their initial stable of games, but he decided to go with another publisher.

In any case, our story continues with this man, Bill Budge, in the top right.

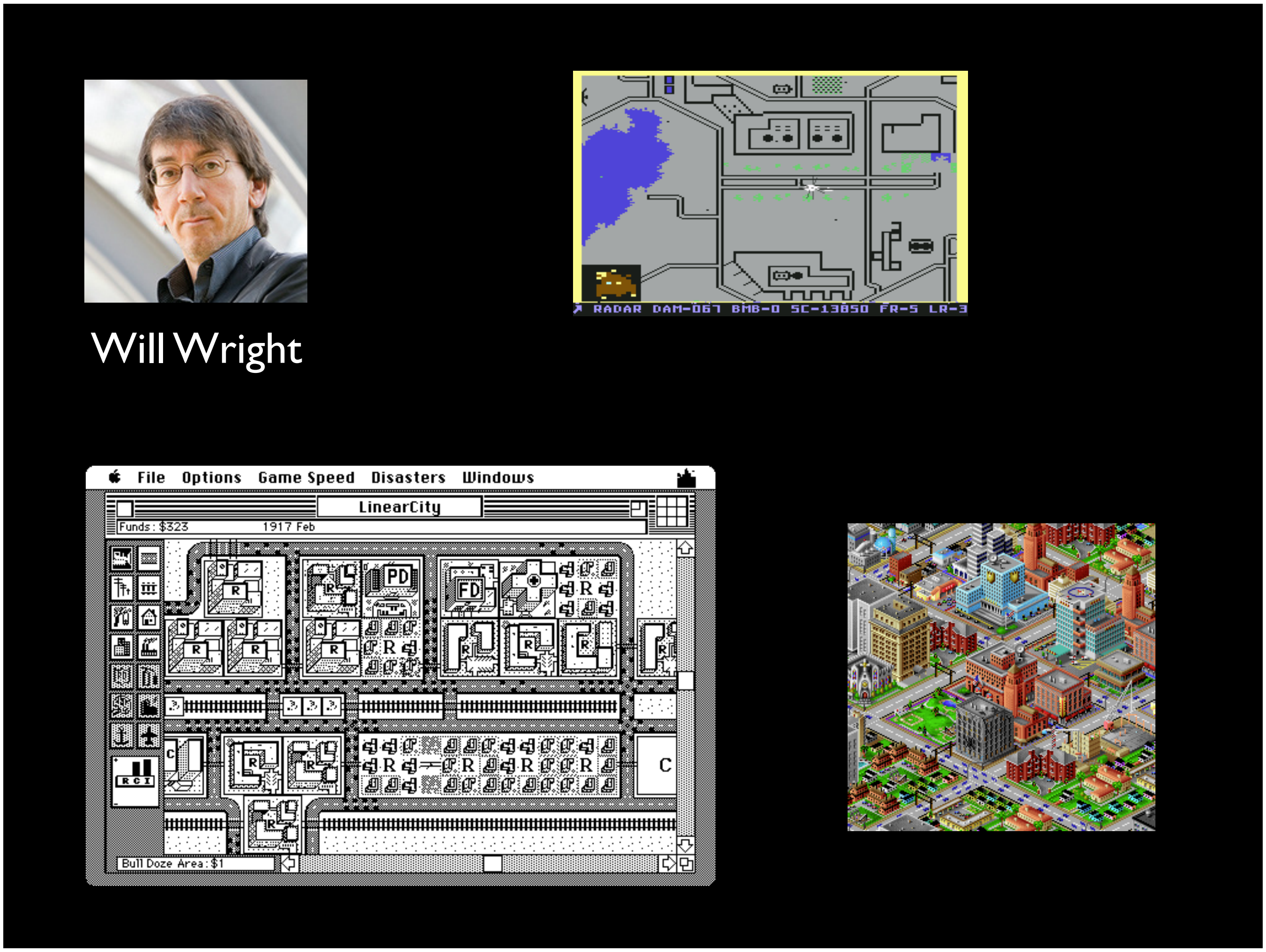


1983, Bill Budge's PCS is released. You design pinball machines—and play them.

It augmented the humans's ability to design and play with pinball machines.

<click> The rhetoric of the ad is fantastic: man/machine interface—"a nearly telepathic link between you and the machine"—a popular marketing echo of Bush and Engelbart's call for a computer to augment and extend the human's creative ability. [power—violating the laws of gravity, Pinball. "Thumb your nose at Newton."] And EA writes that their goal is to "fulfill the potential of computing"—which sounds an awful lot like the creators' of SpaceWar! criteria that games maximally utilize and show off the capabilities of the machine. "The reason you bought the machine in the first place."

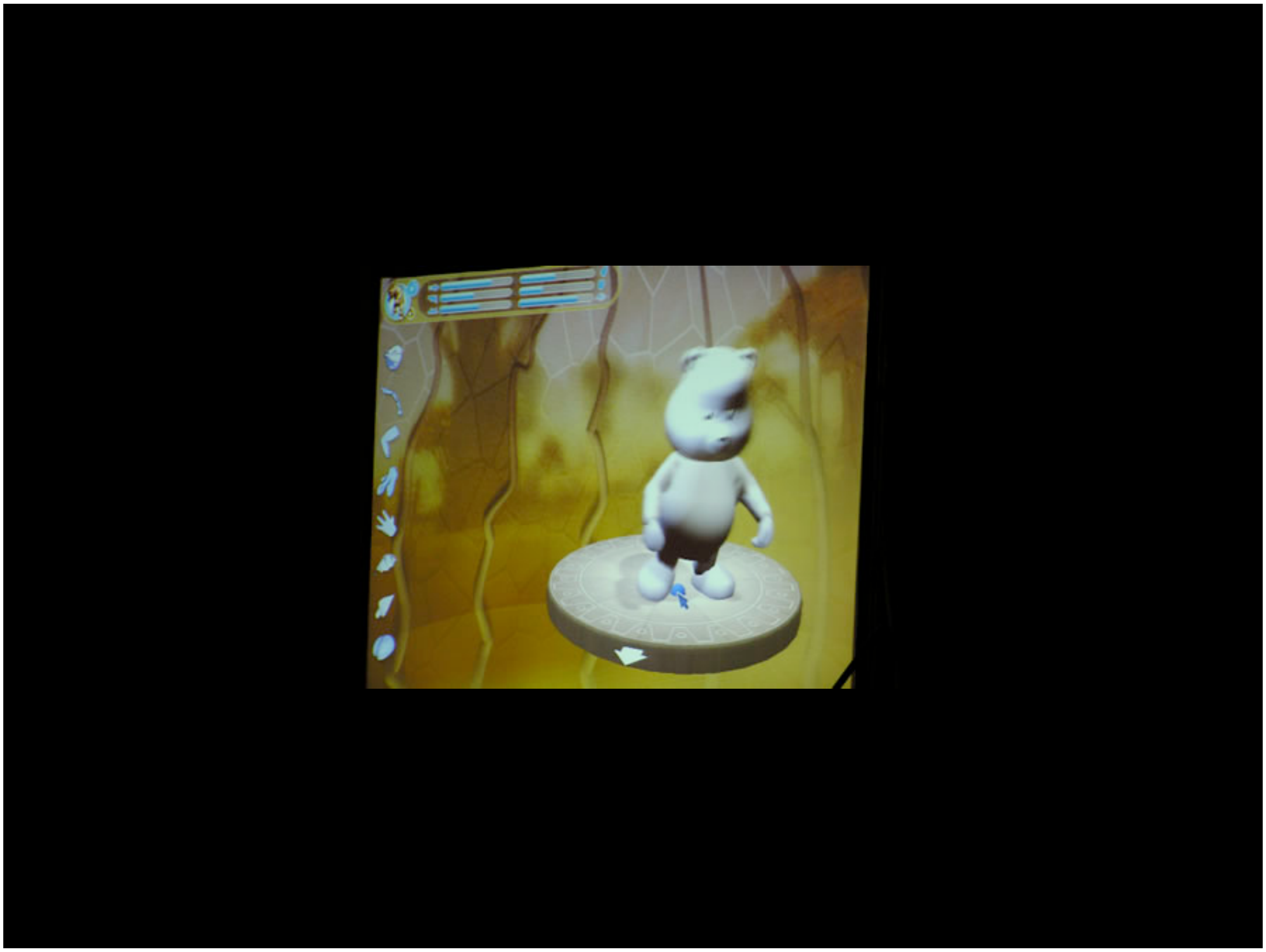
Marketing aside, this was a landmark game—if we can call it that, really, since while it is a pinball game, it is also something more: a toy, a construction set. It used a direct manipulation style interface, analogous to the one the Xerox Star used. In a sense, PCS harks back to the pre-SpaceWar era, when Graetz and his friends were exploring how to apply the playful possibilities of the machine, before they hit upon the idea of making a game. PCS is play with a computer that isn't quite a computer game, and it points the way towards something else.

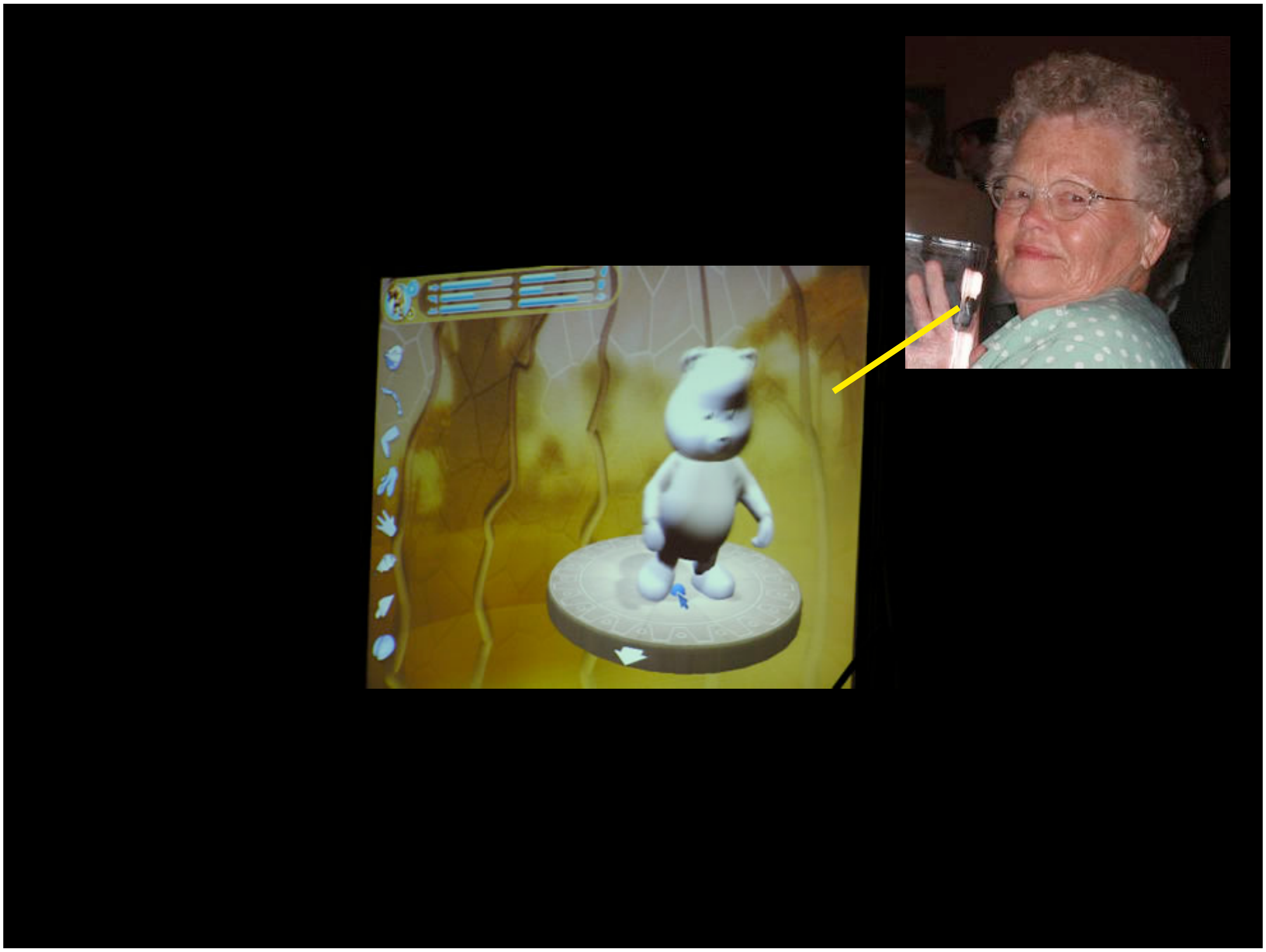


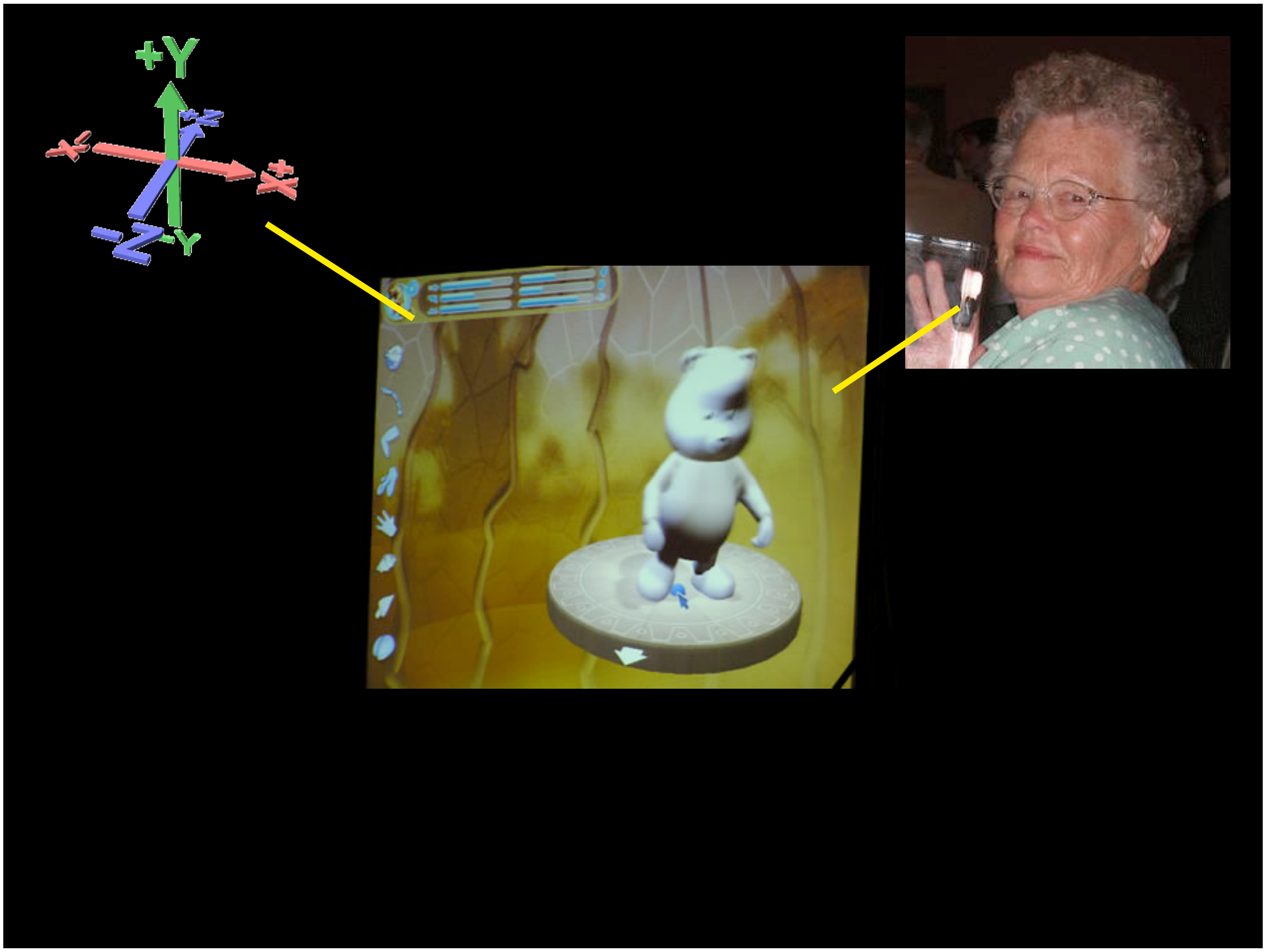
In 1984, Will Wright's first game—Raid on Bunging Bay, is published. The player pilots a fighter helicopter against a (both fictional and real) AI controlled complex of industrial and military facilities.

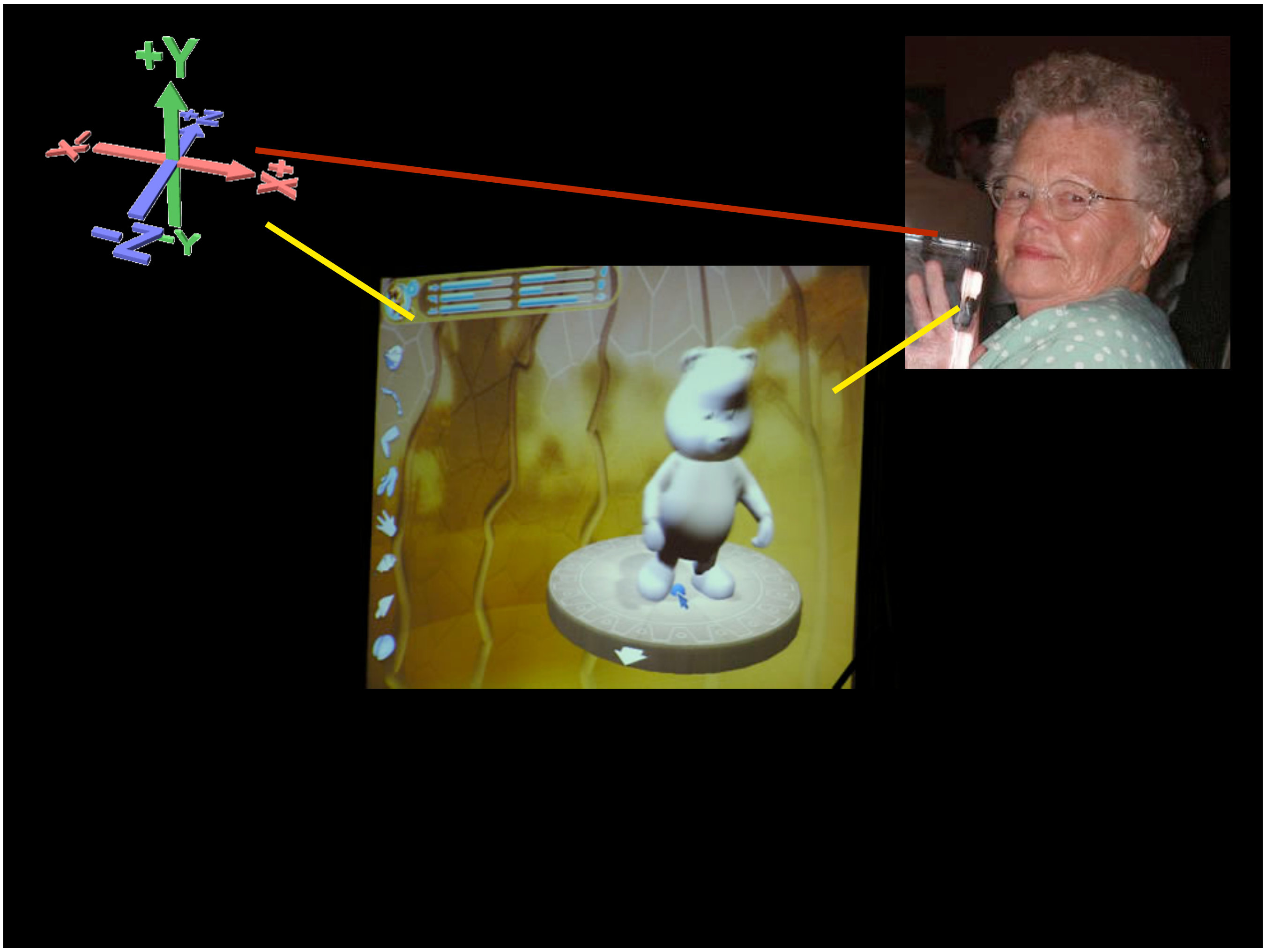
Will, though, finds that he has more fun in the map editor–designing cityscapes to blow up, then he does blowing them up. After playing Pinball Cons. Set, he realizes that he could make this—the joy of creating something—into a product, a computer toy.

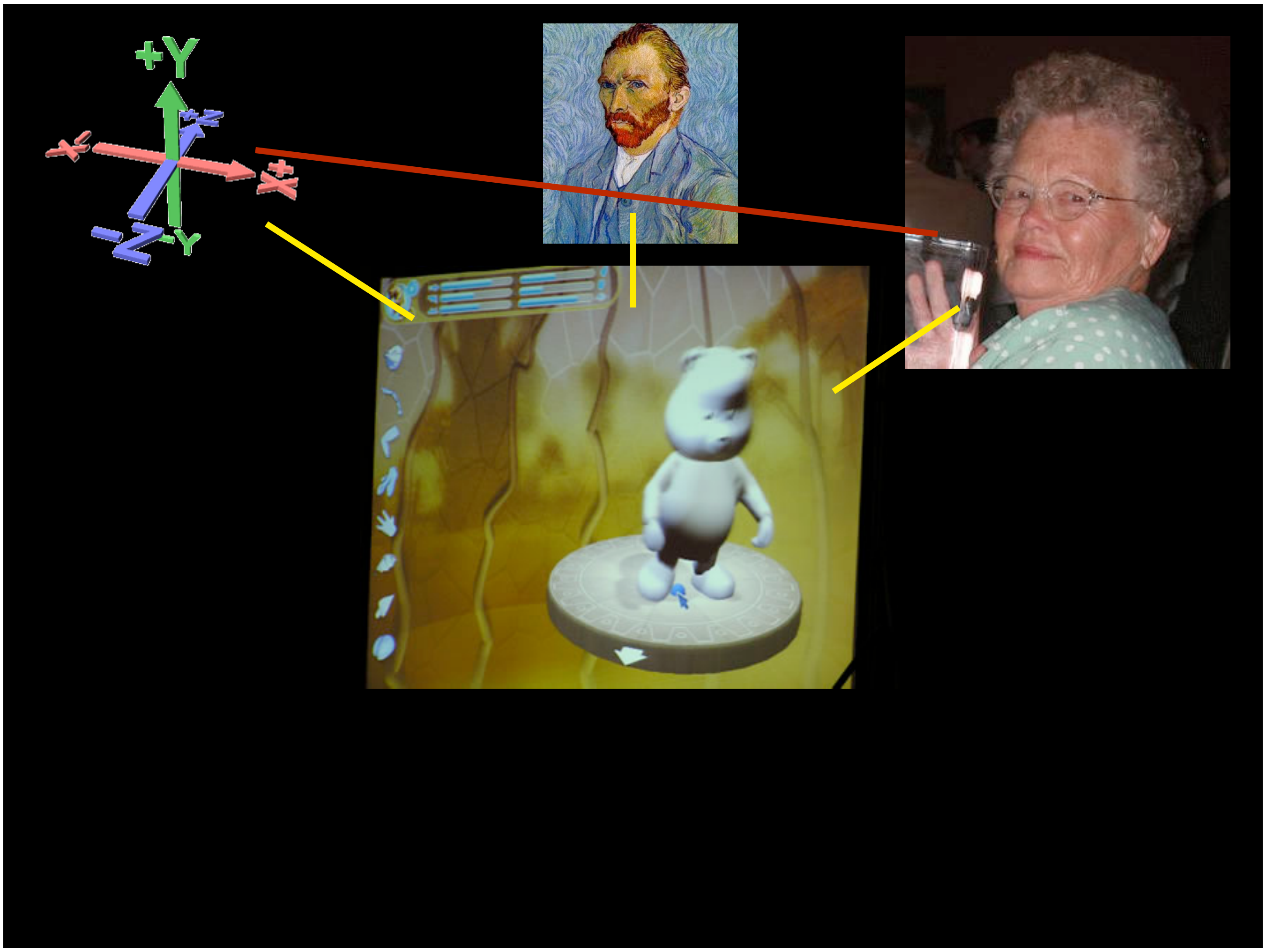
Will works on this for about 5 years, incorporating Jay Forrester's research on city simulation, and cellular automata, and creates SimCity <click> which again establishes a whole new set of conventions and expectations for playful computer software "games"—that aren't really games at all.

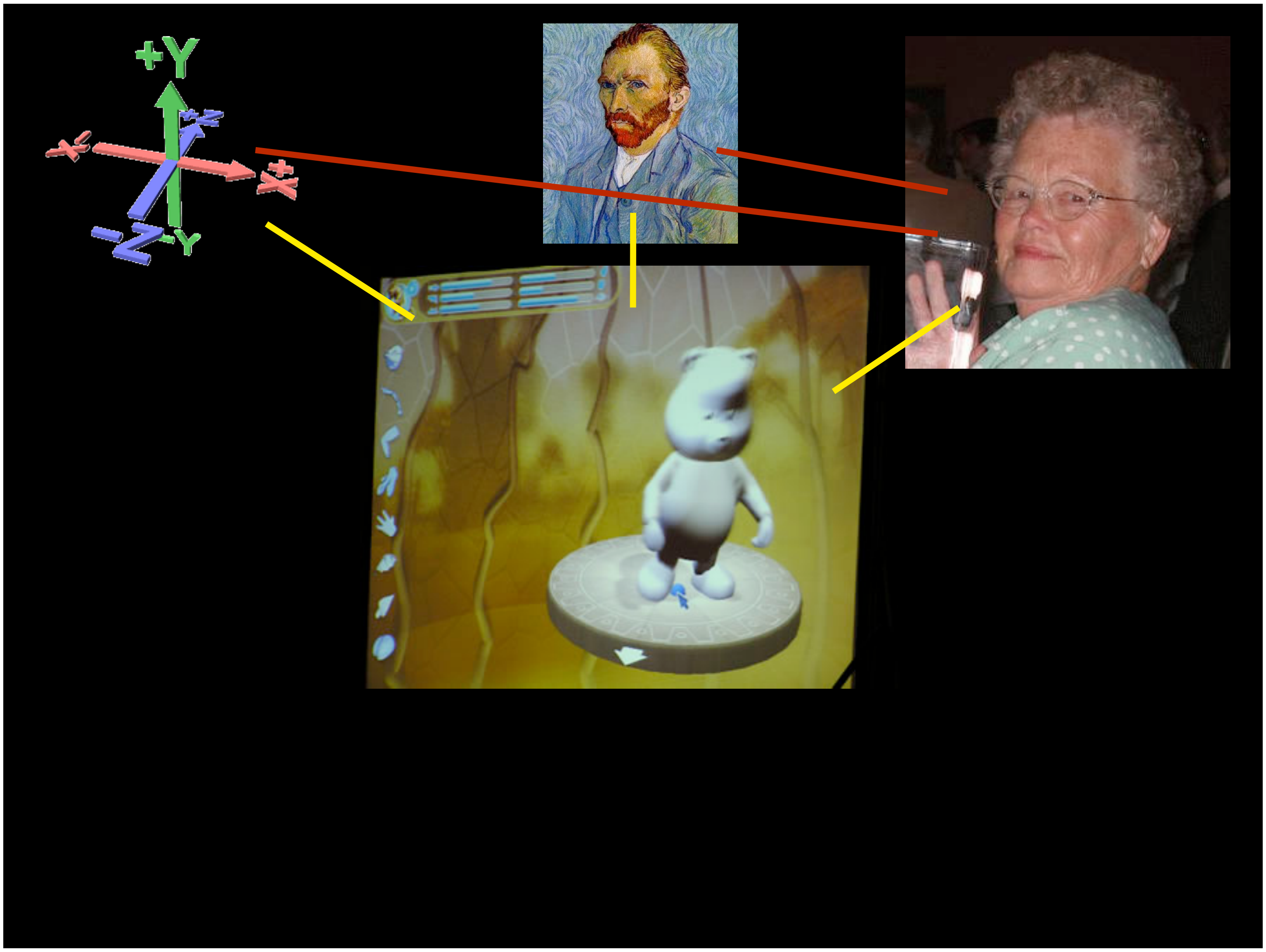


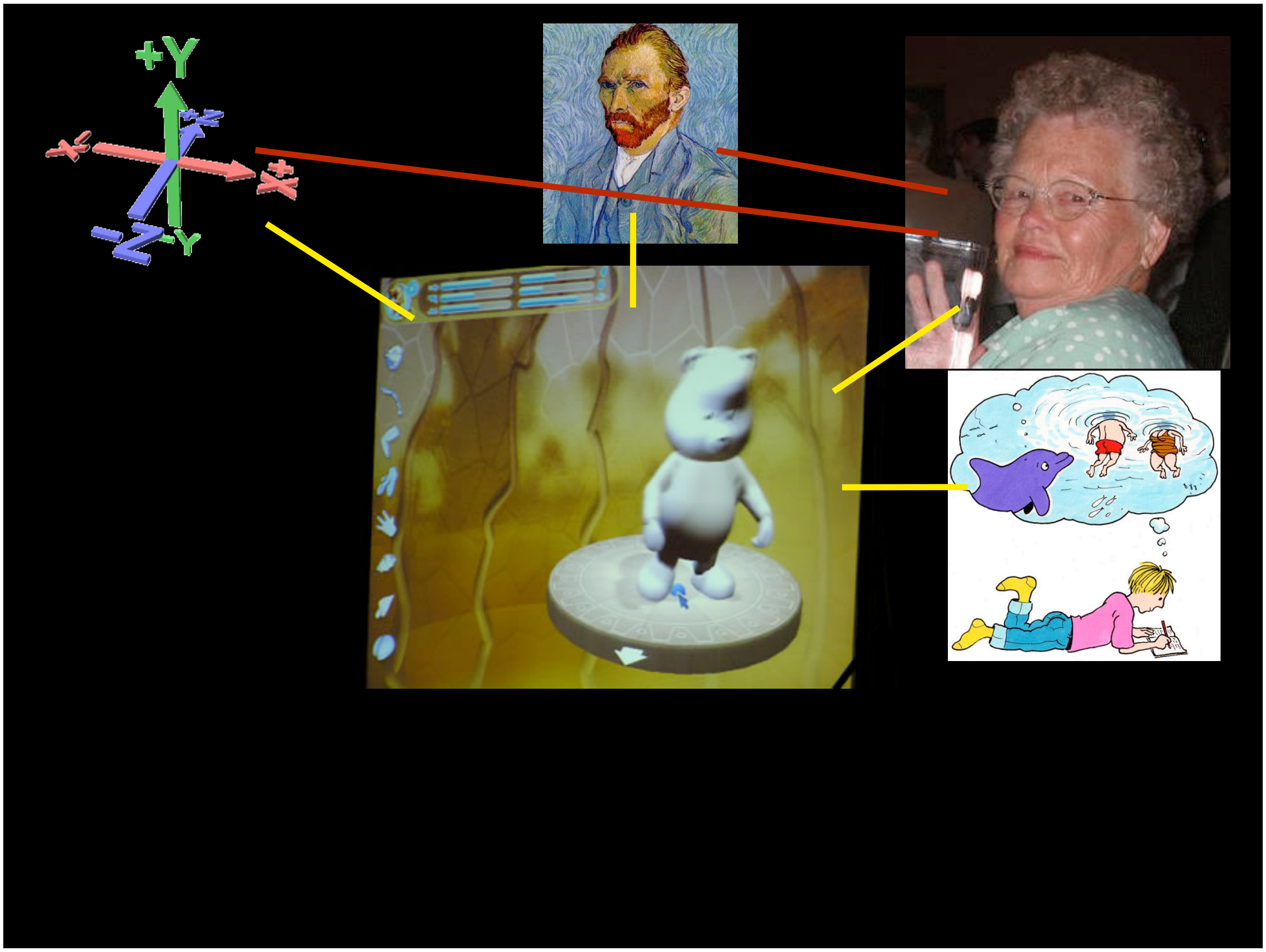


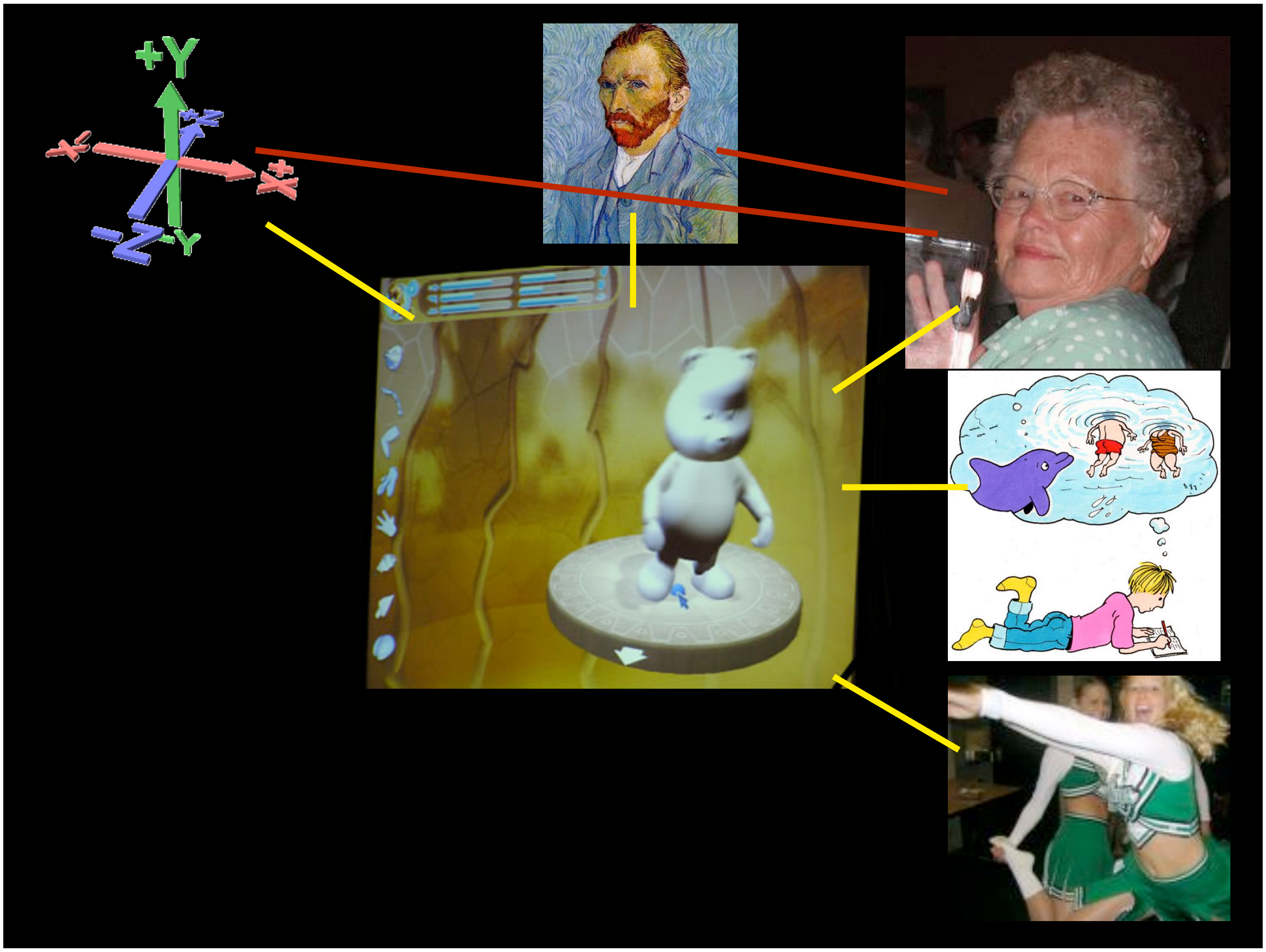

+Y
-Y
-Z

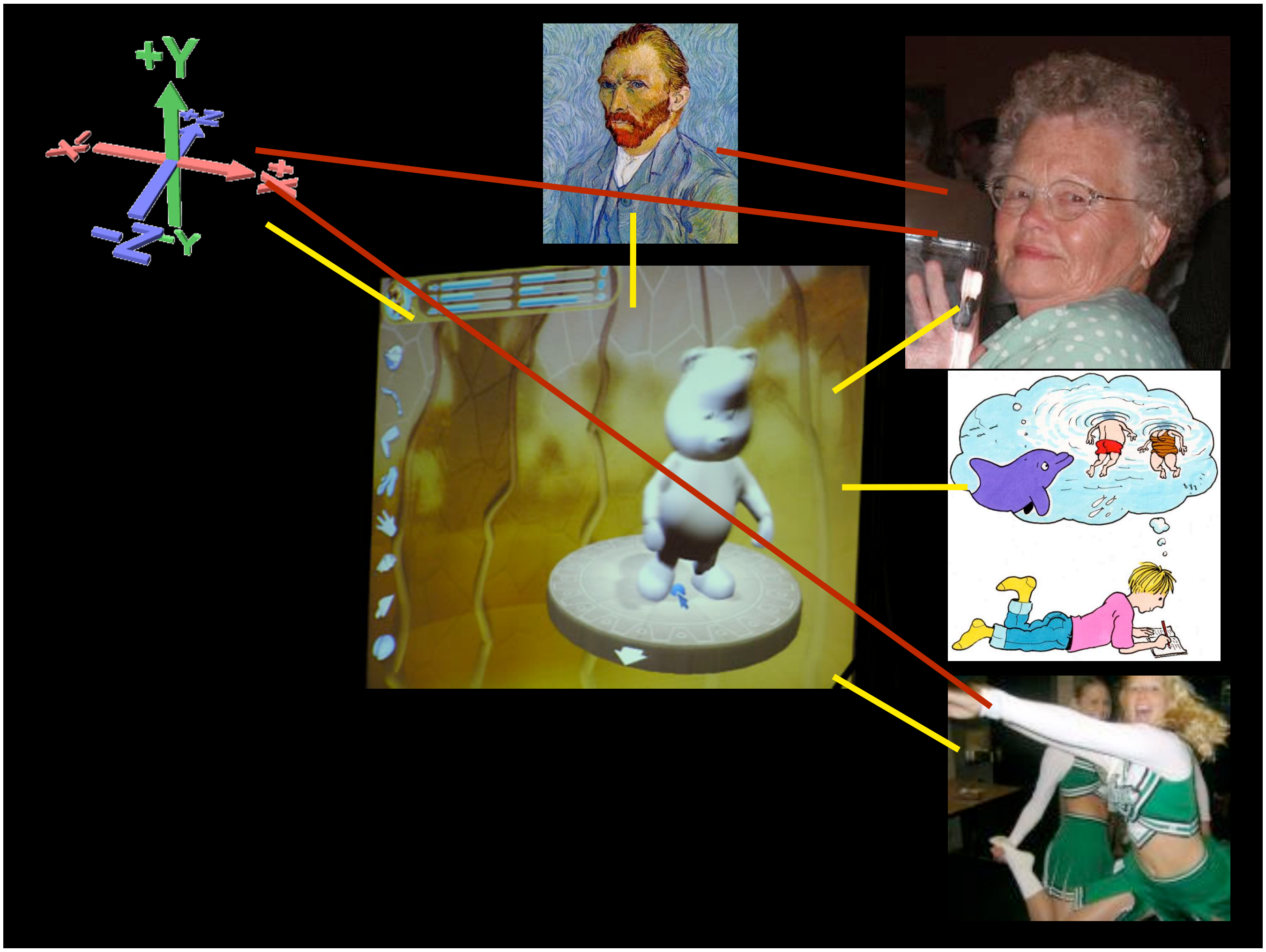

+Y
+X
-X
-Y
-Z
+Z


+Y
-X
+Z
+X
-Z
-Y

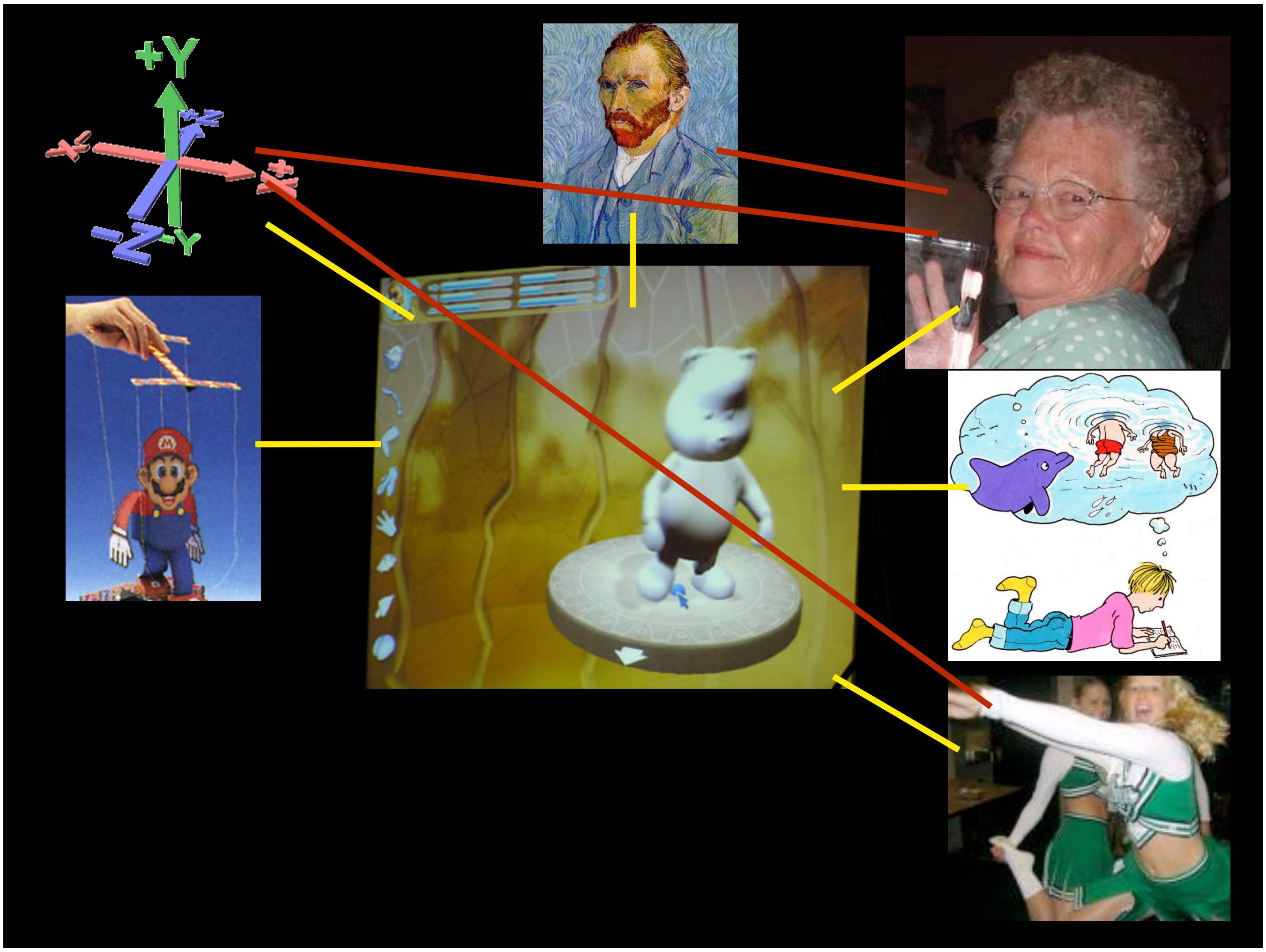

+Y
-X
+X
+Z
-Z
-Y

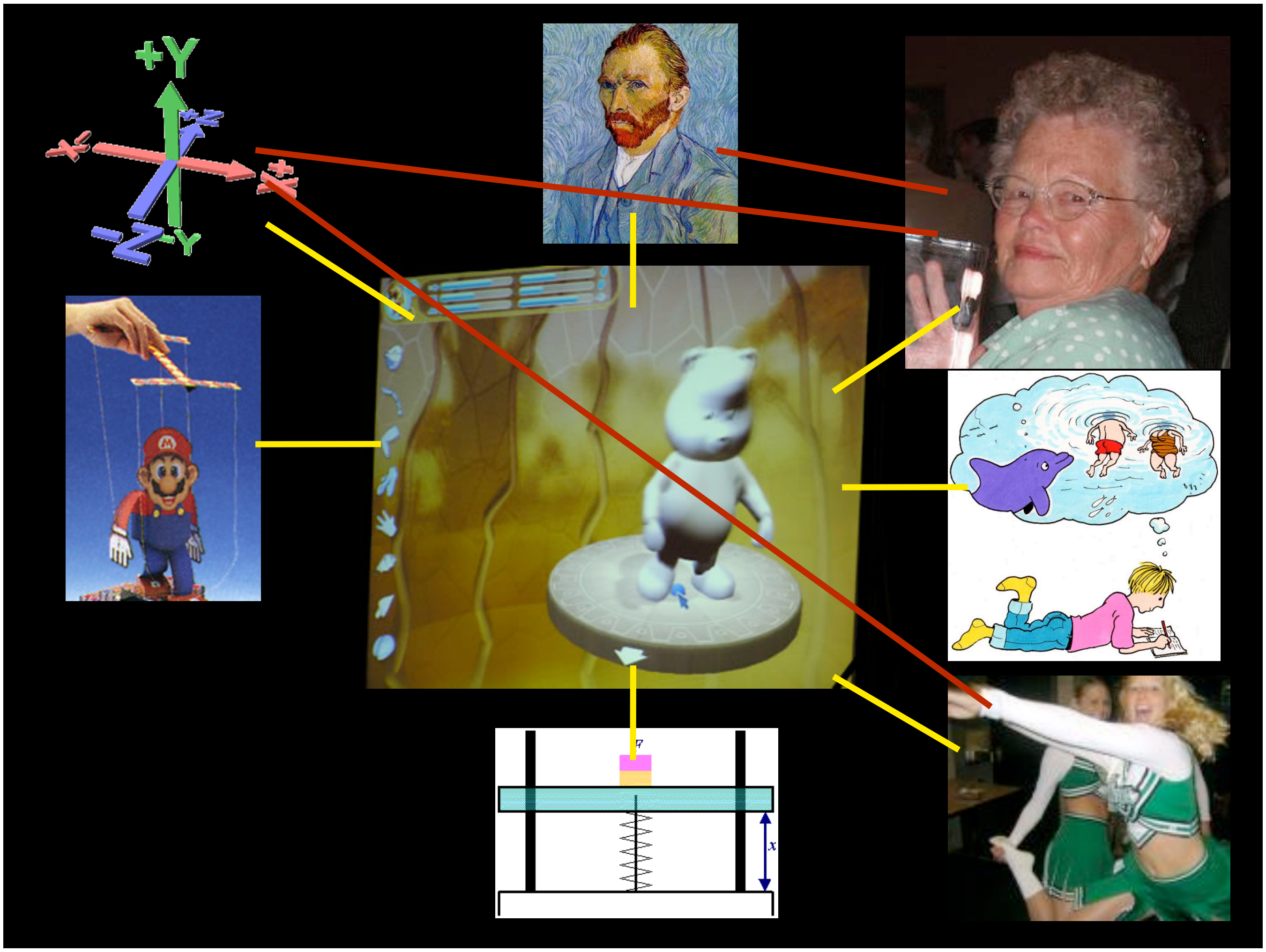

+Y
-Y
+X
-X
+Z
-Z


+Y
-X
+X
+Z
-Z
-Y

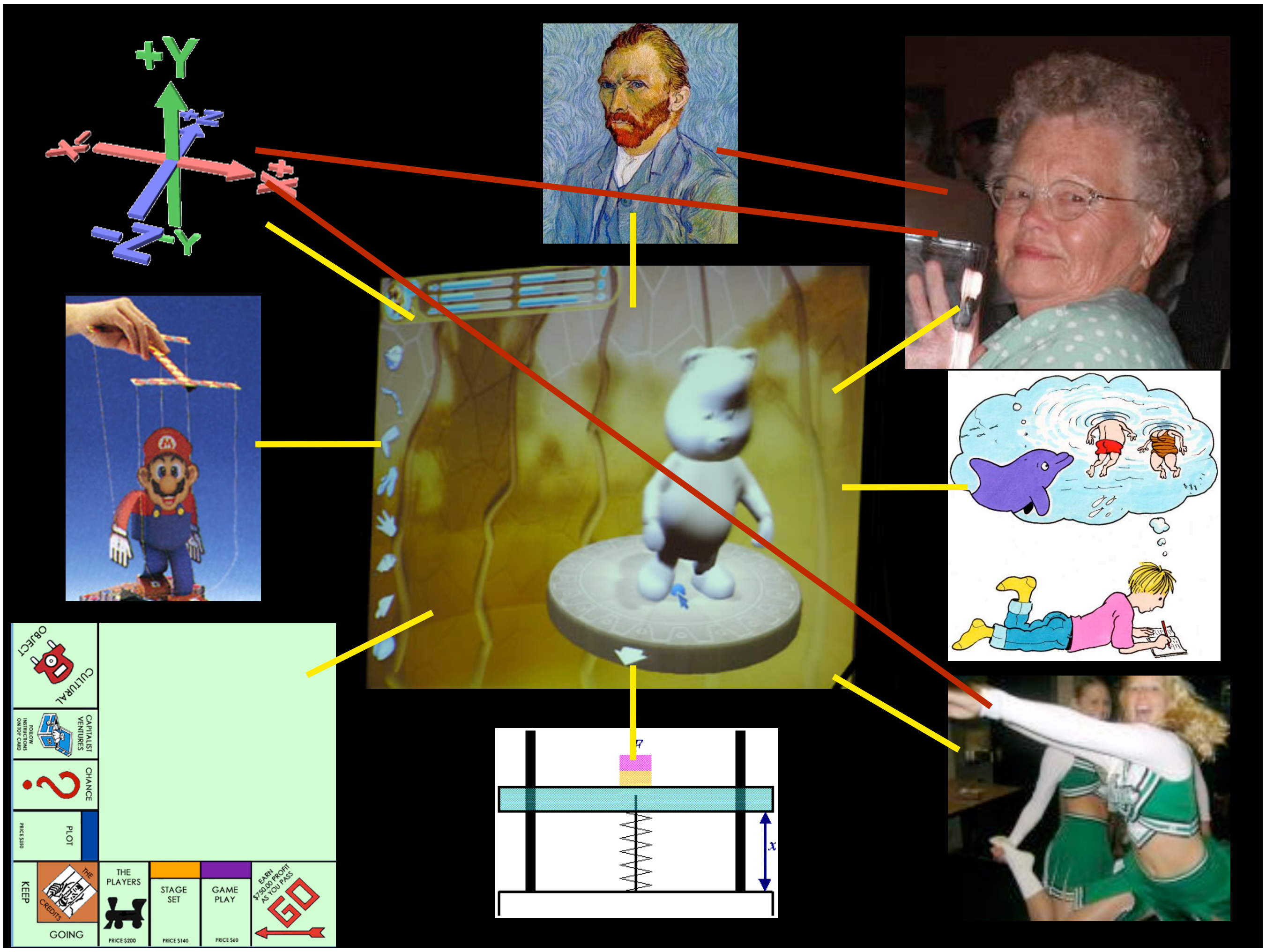


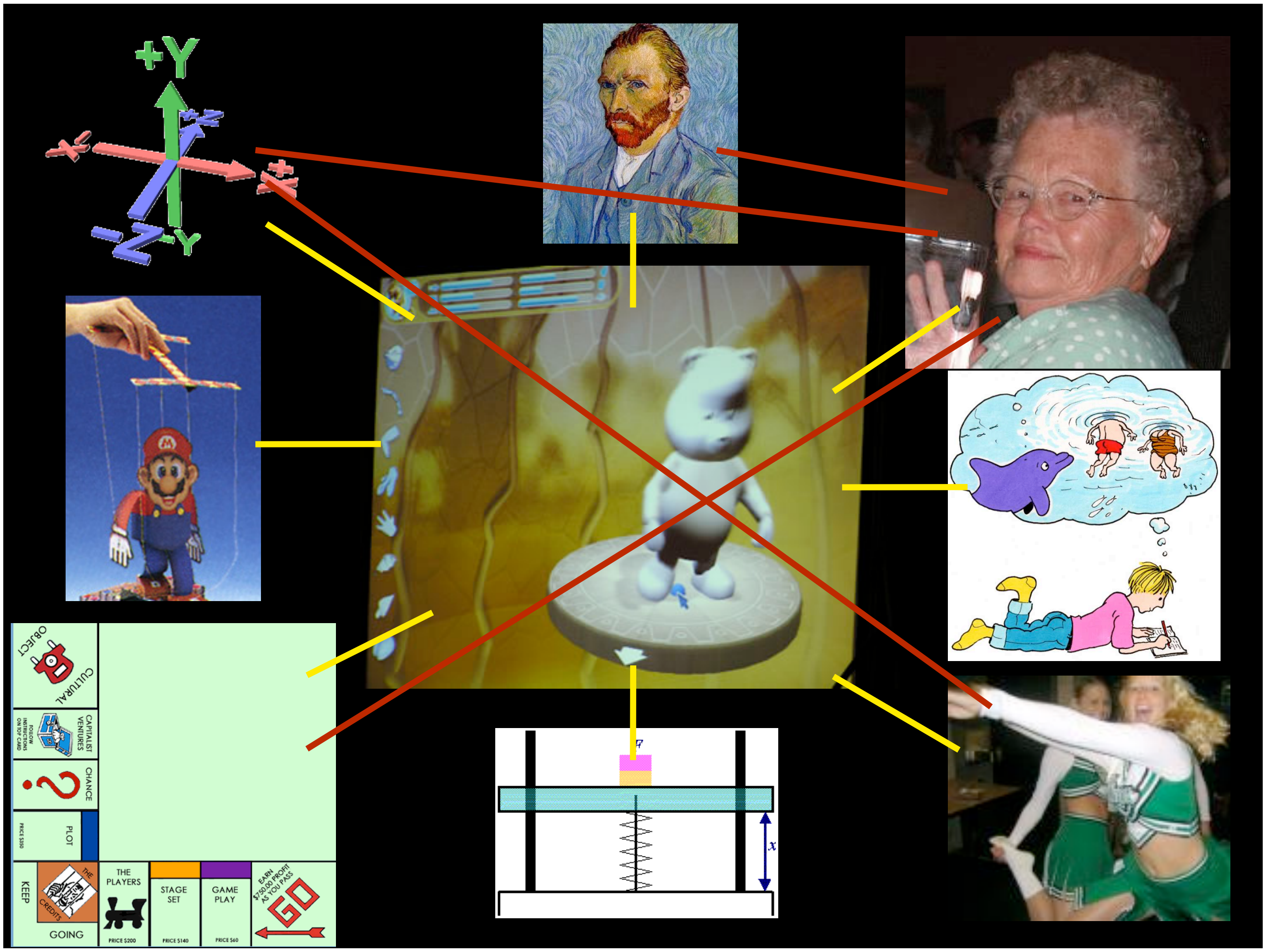
+Y
-X
+X
+Z
-Z
-Y

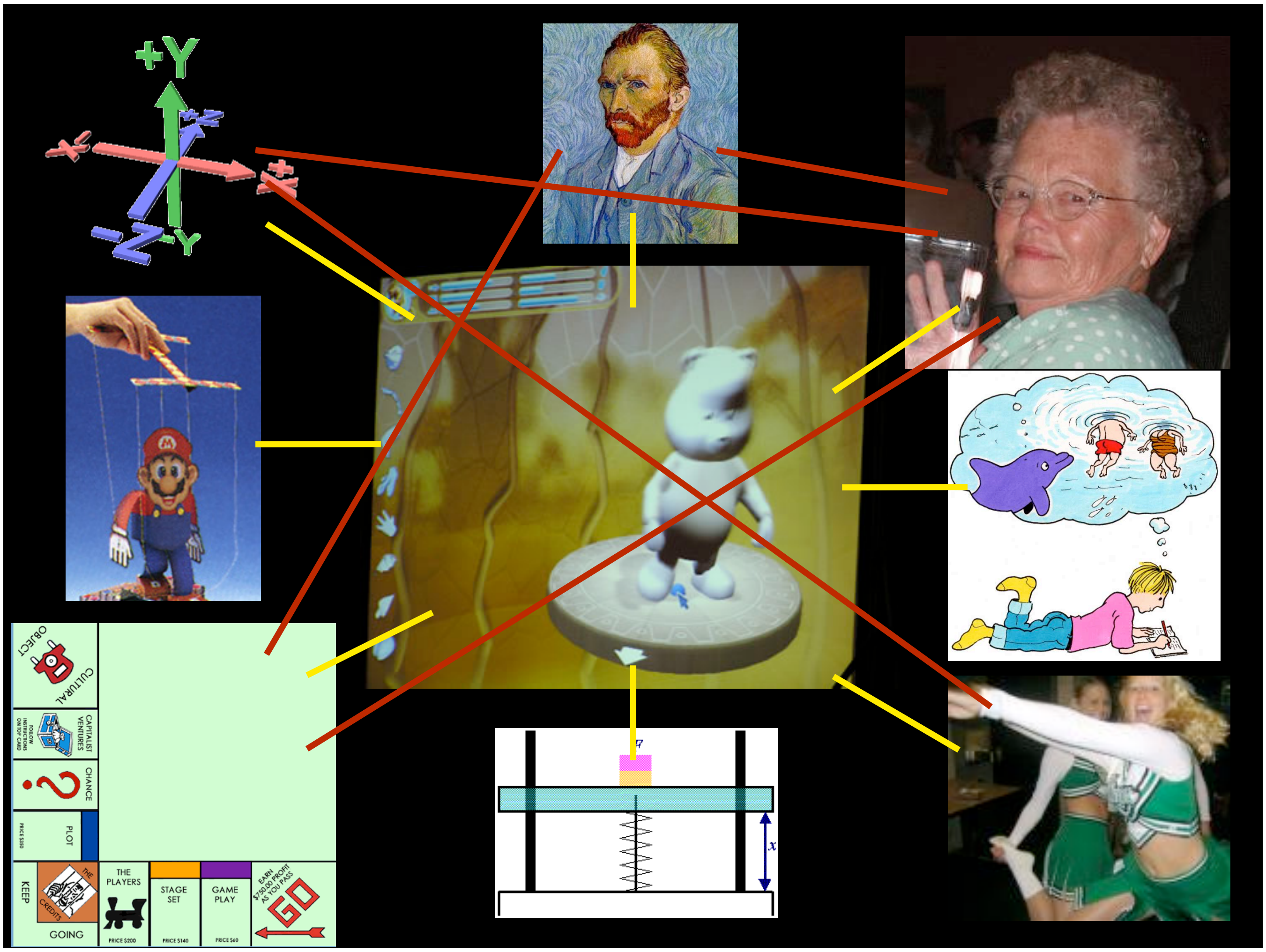

+Y
-Z
-Y



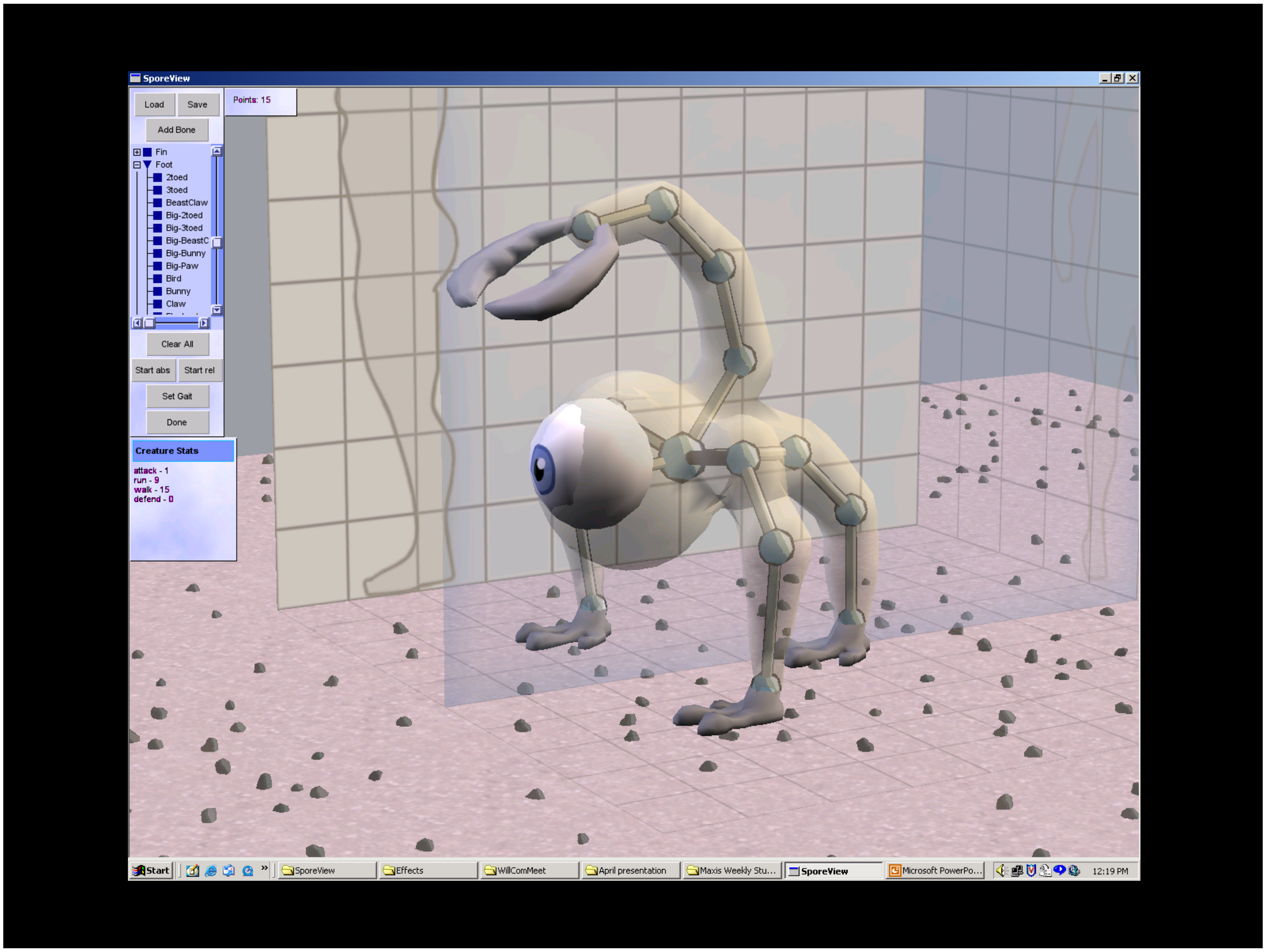
+Y
-Z
x

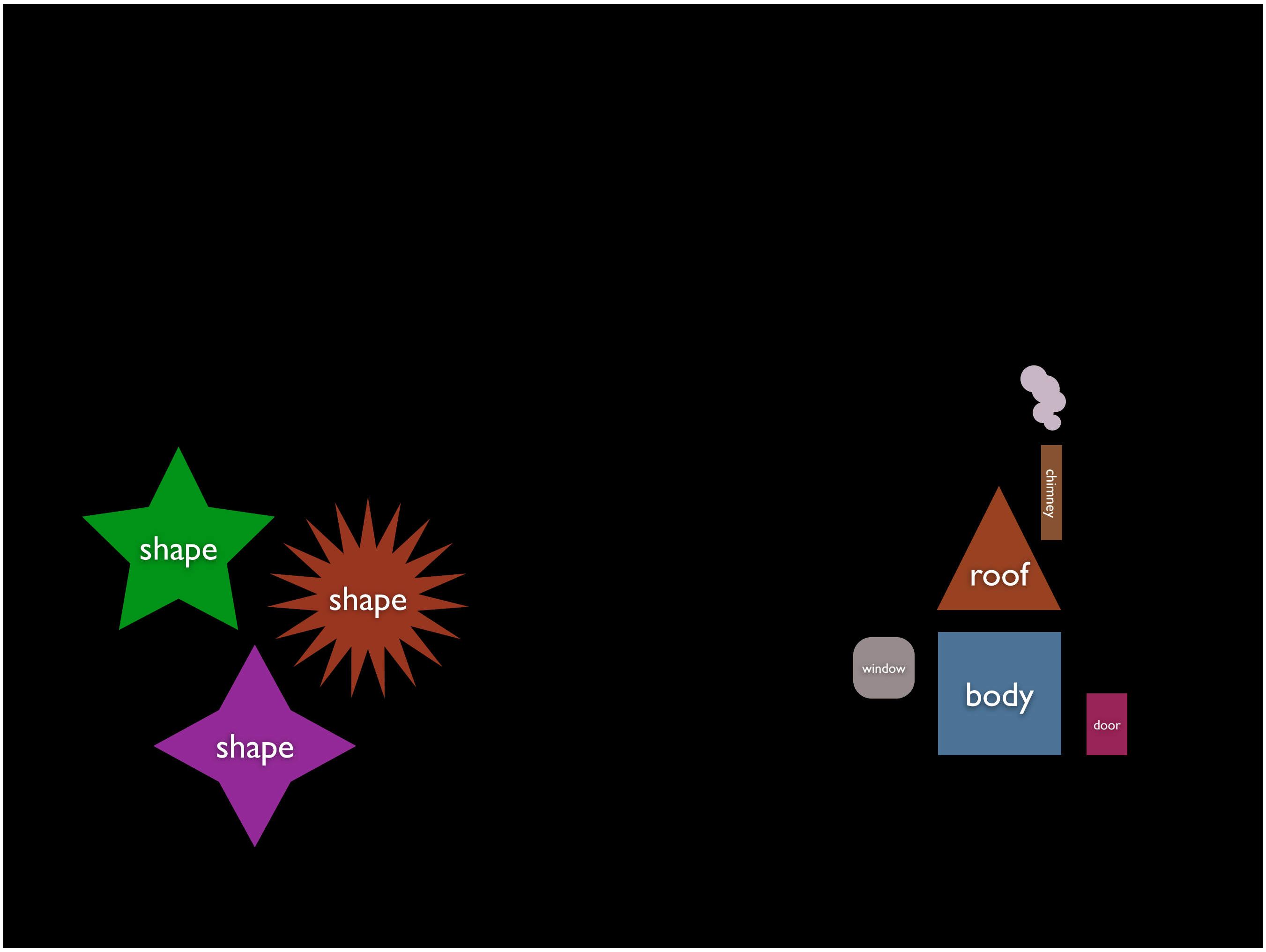

+Y
-Y
-Z
CULTURAL OBJECT
CAPITALIST VENTURES
FOLLOW INSTRUCTIONS ON TOP CARD
CHANCE
PLOT
PRICE $350
KEEP
THE CREDITS
GOING
THE PLAYERS
PRICE $200
STAGE SET
PRICE $140
GAME PLAY
PRICE $60
EARN $750.00 PROFIT AS YOU PASS
GO
x

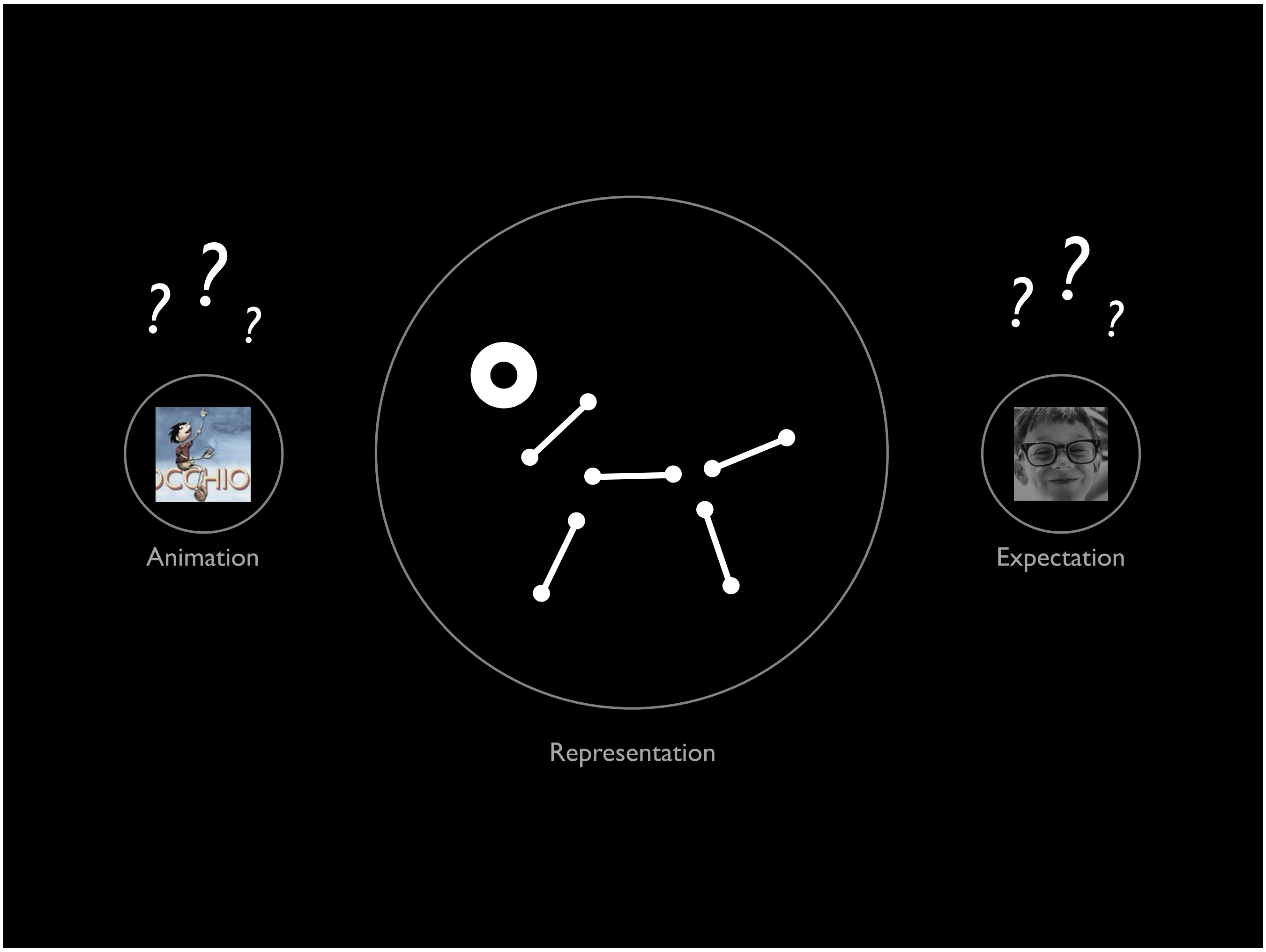

+Y
-Y
-Z
CULTURAL OBJECT
CAPITALIST VENTURES
CHANCE
PLOT
KEEP
THE CREDITS
GOING
THE PLAYERS
PRICE $200
STAGE SET
PRICE $140
GAME PLAY
PRICE $60
GO
x


+Y
-Y
-X
+X
-Z
OBJECT
CULTURAL
CAPITALIST VENTURES
CHANCE
PLOT
KEEP
THE CREDITS
GOING
THE PLAYERS
PRICE $200
STAGE SET
PRICE $140
GAME PLAY
PRICE $60
EARN $750.00 PROFIT AS YOU PASS
GO
x

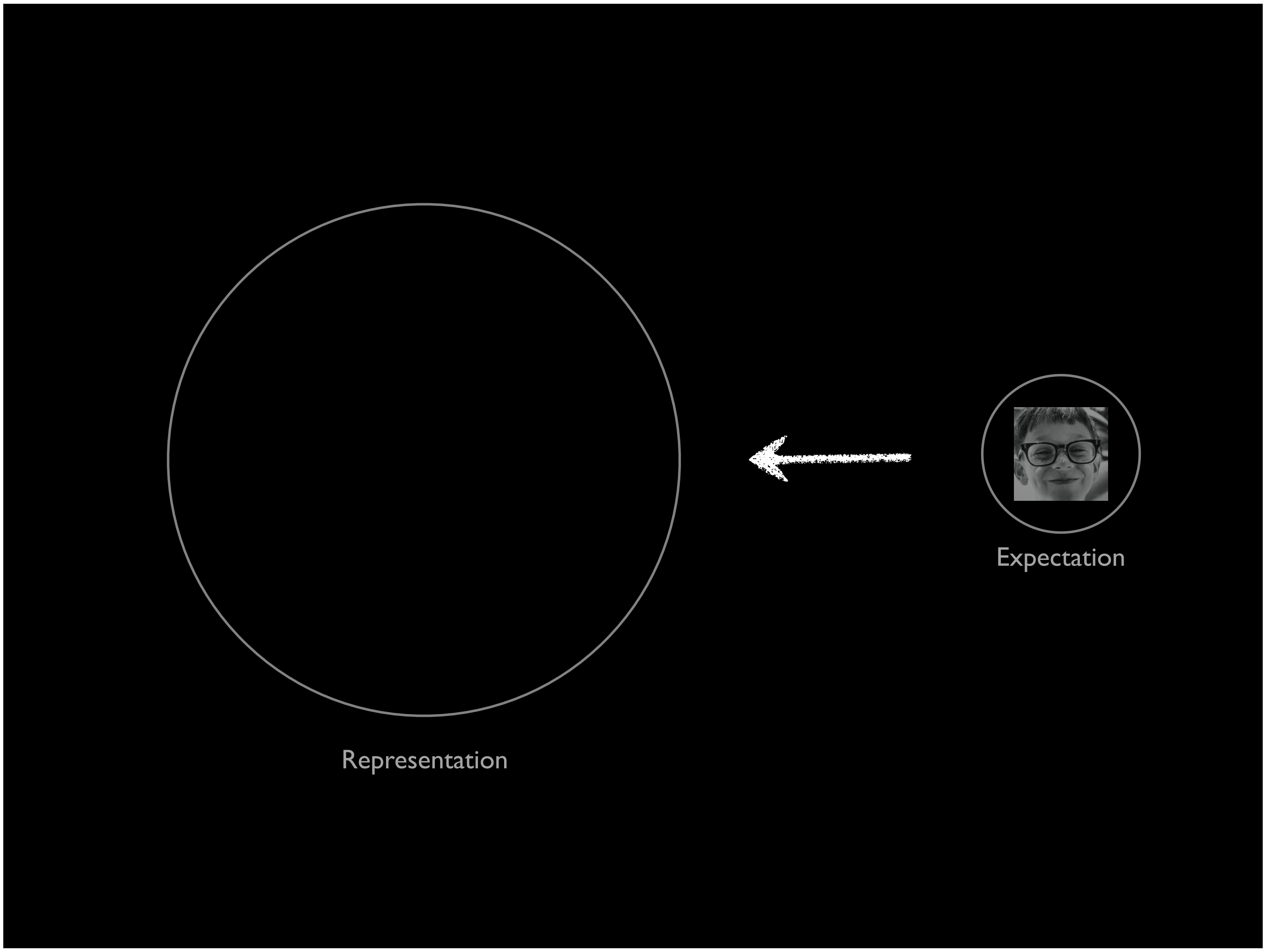
Deep Structure

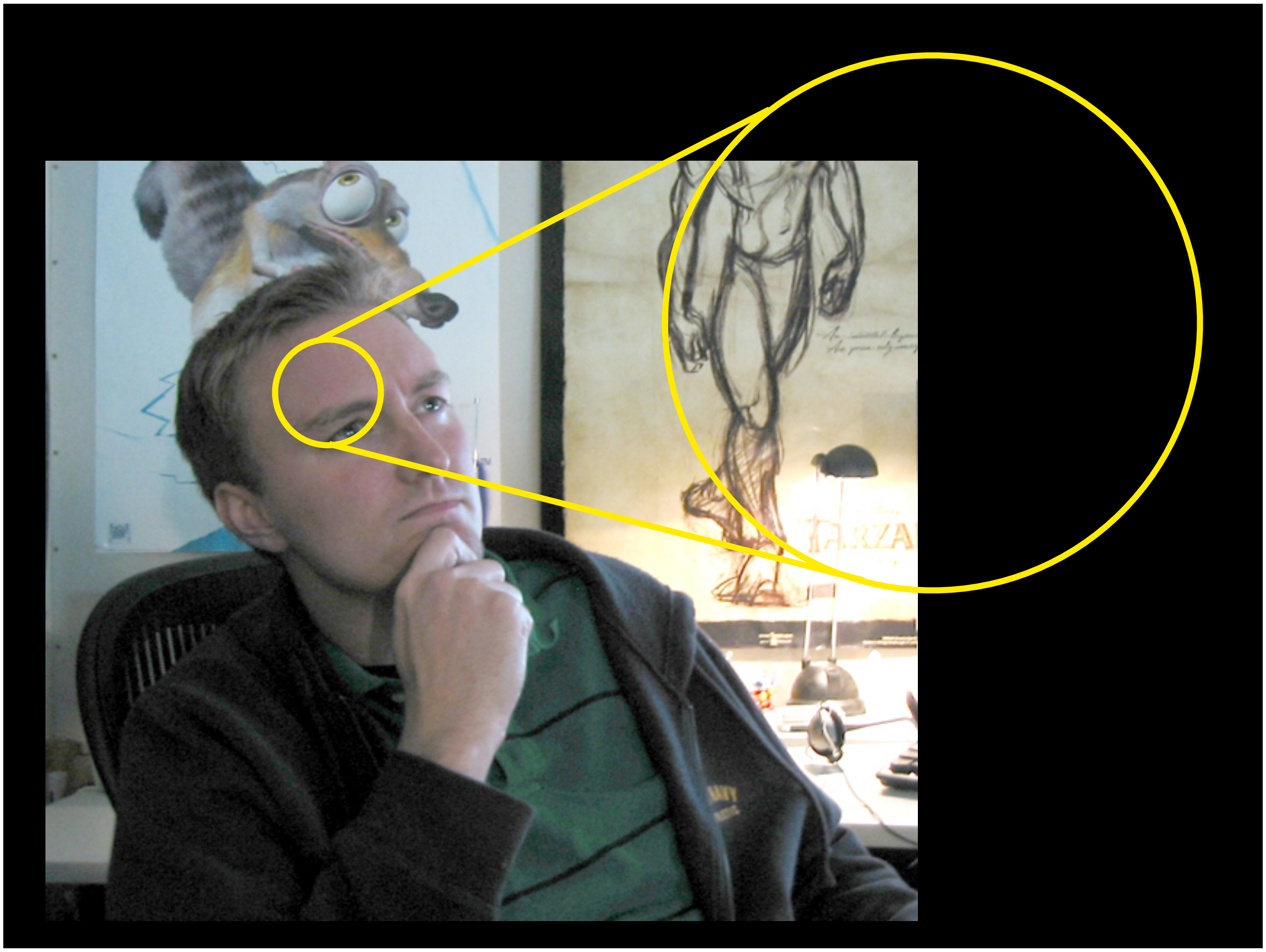

SporeView
Load
Save
Add Bone
Points: 15
Fin
Foot
2toed
3toed
BeastClaw
Big-2toed
Big-3toed
Big-BeastC
Big-Bunny
Big-Paw
Bird
Bunny
Claw
Clear All
Start abs
Start rel
Set Gait
Done
Creature Stats
attack - 1
run - 9
walk - 15
defend - 0
Start
SporeView
Effects
WillComMeet
April presentation
Maxis Weekly Stu...
SporeView
Microsoft PowerPo...
12:19 PM

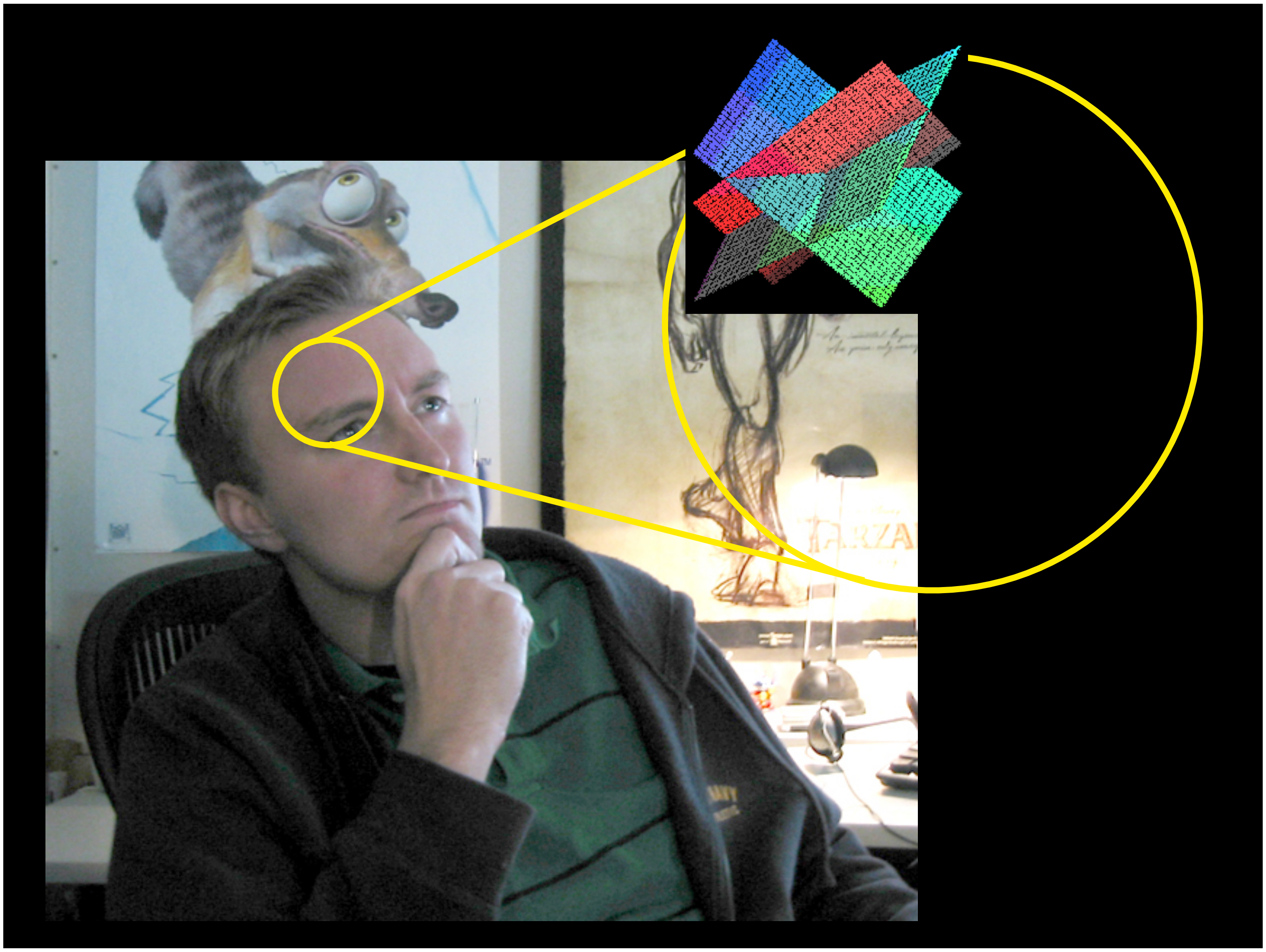

shape
shape
shape
chimney
roof
window
body
door

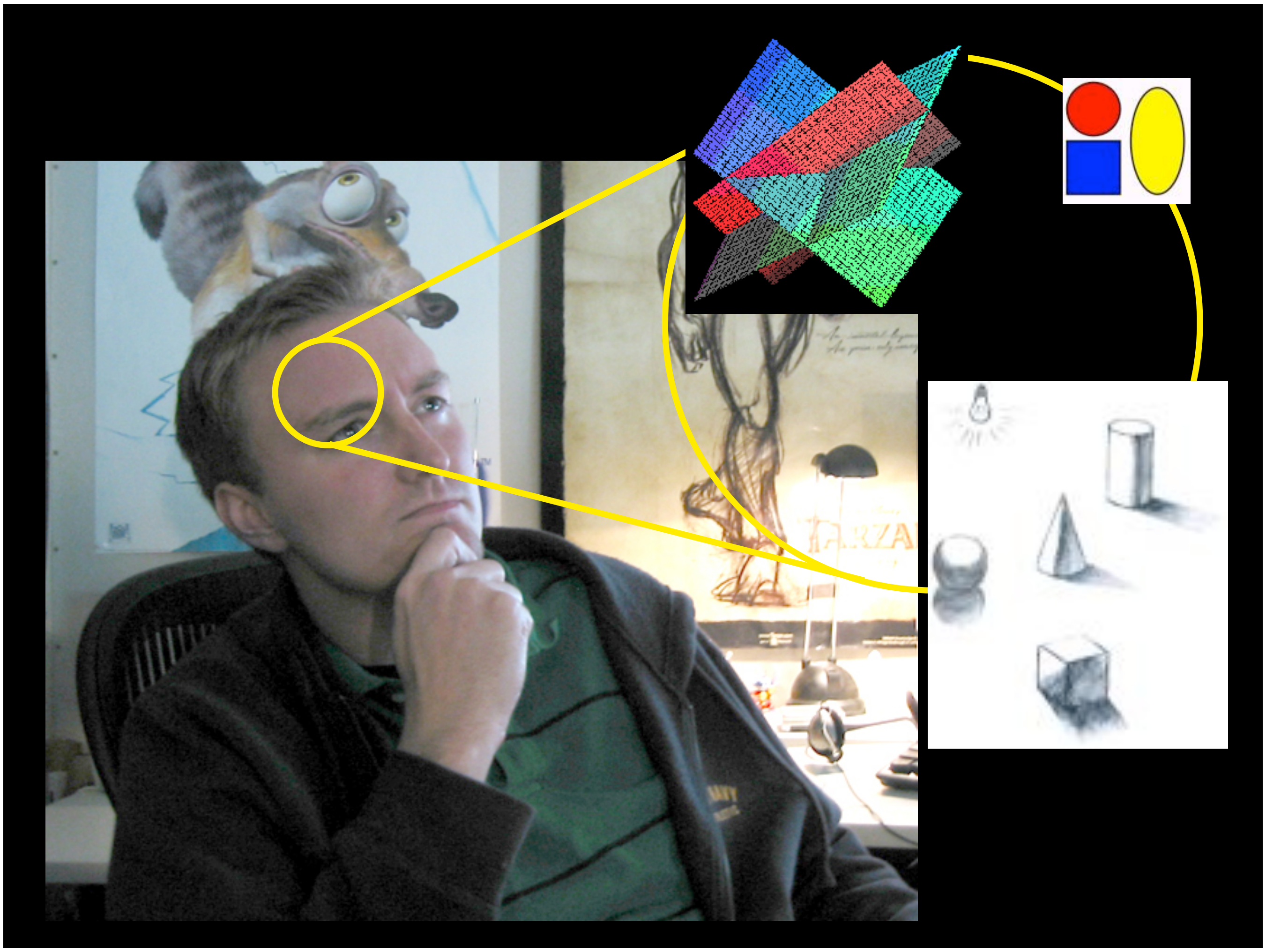


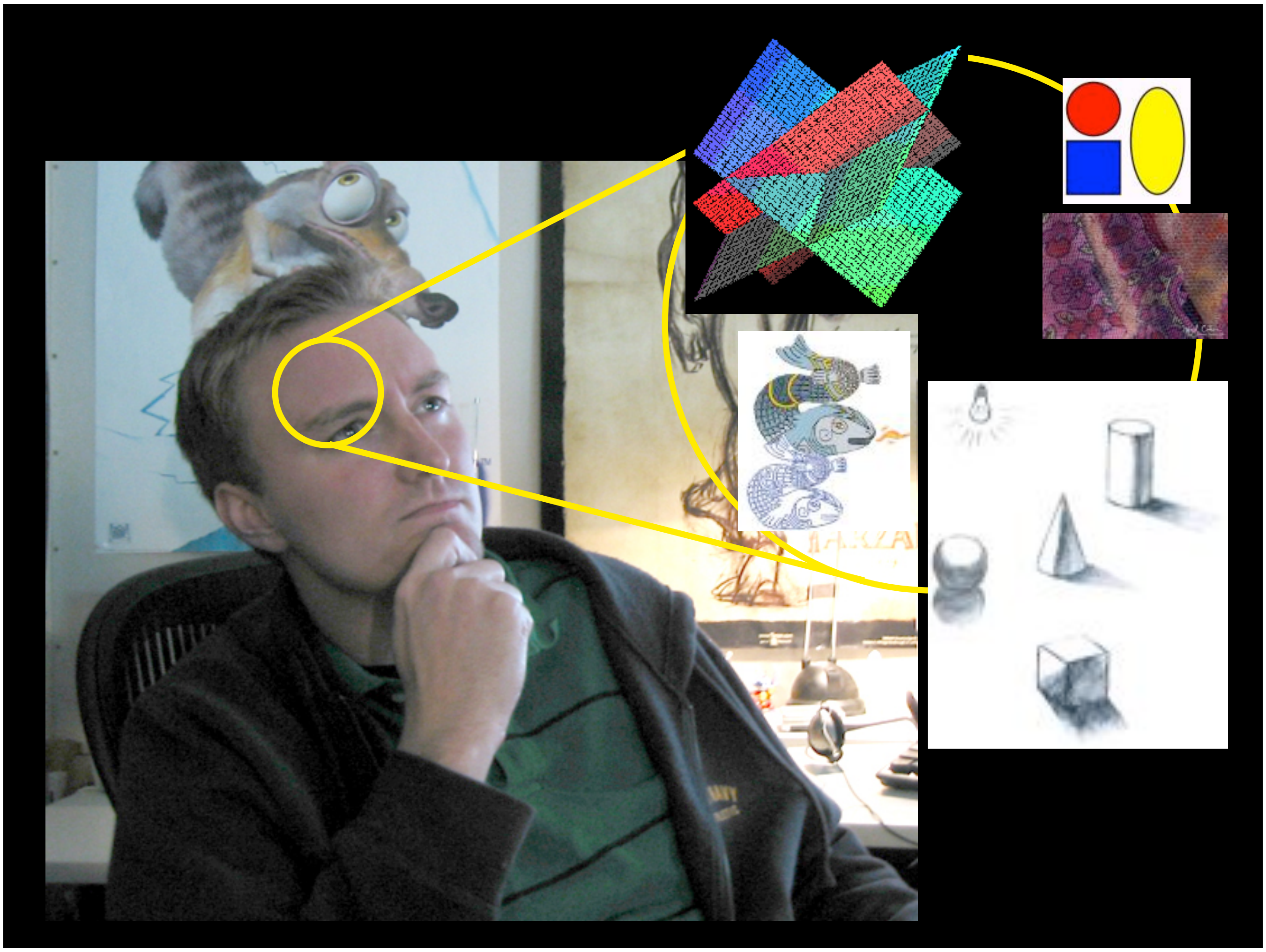
Animation
Representation
Expectation

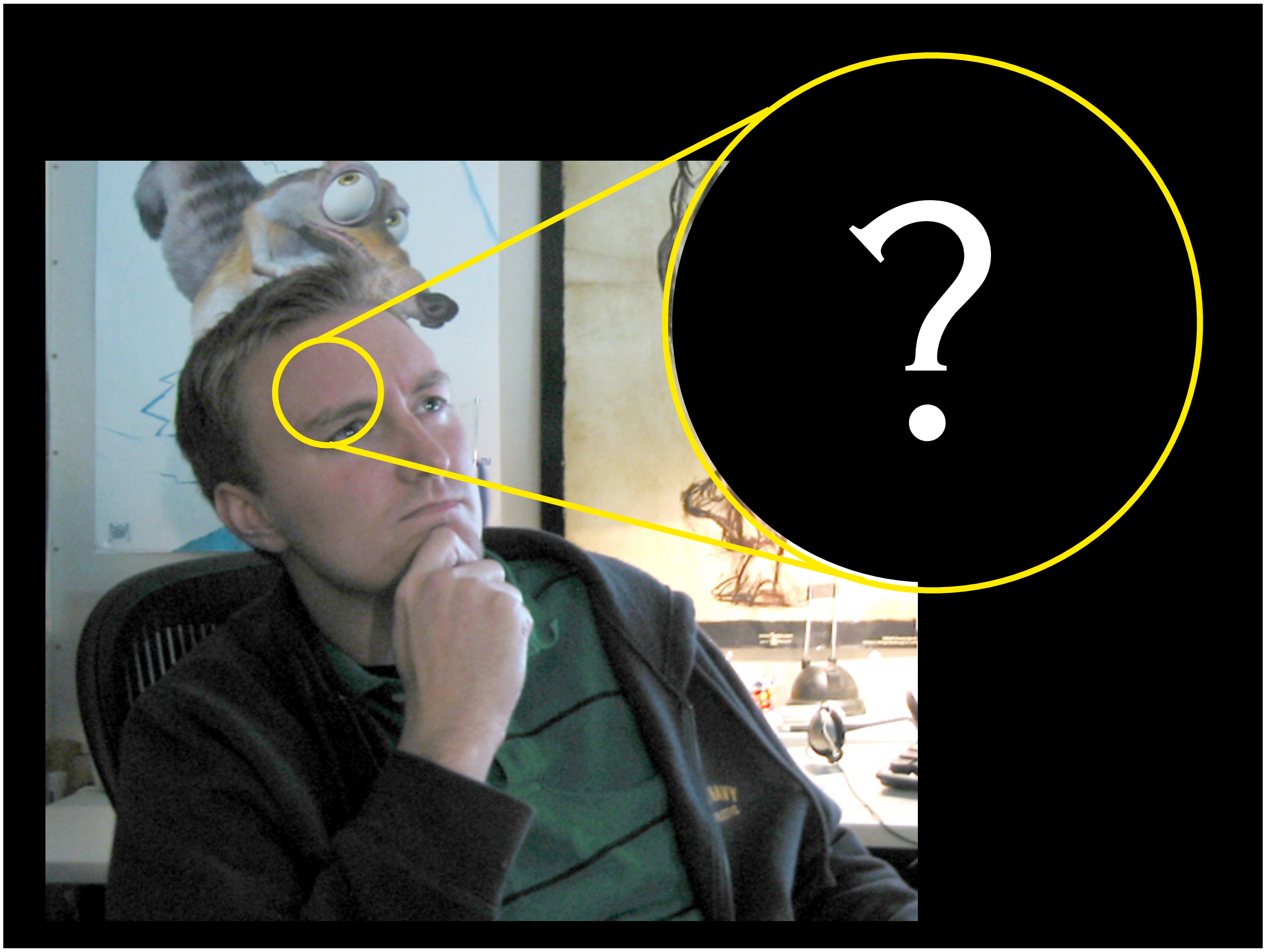

Representation
Expectation

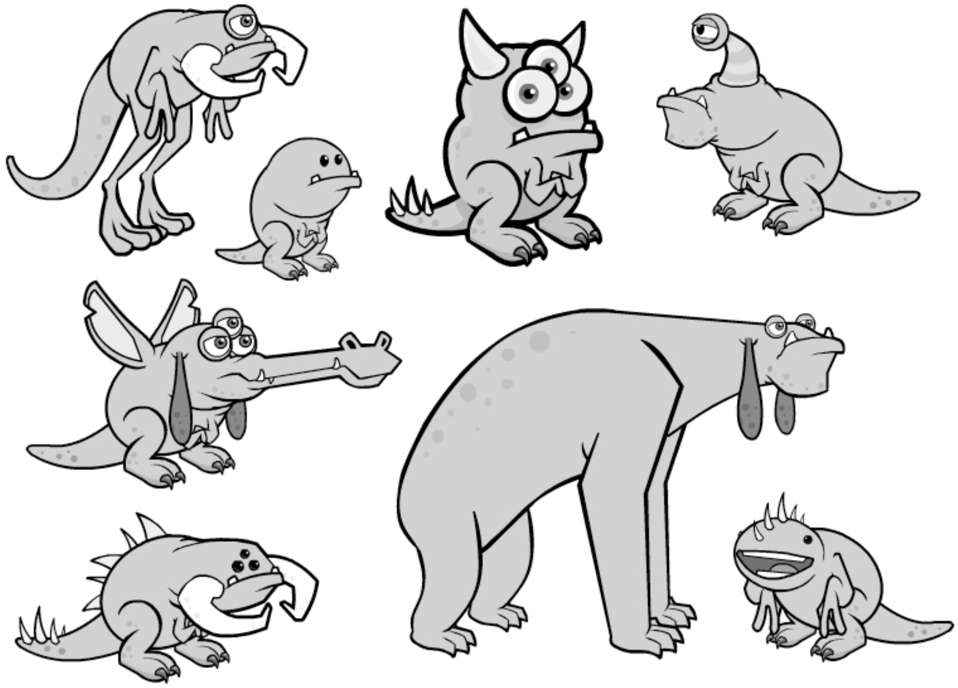


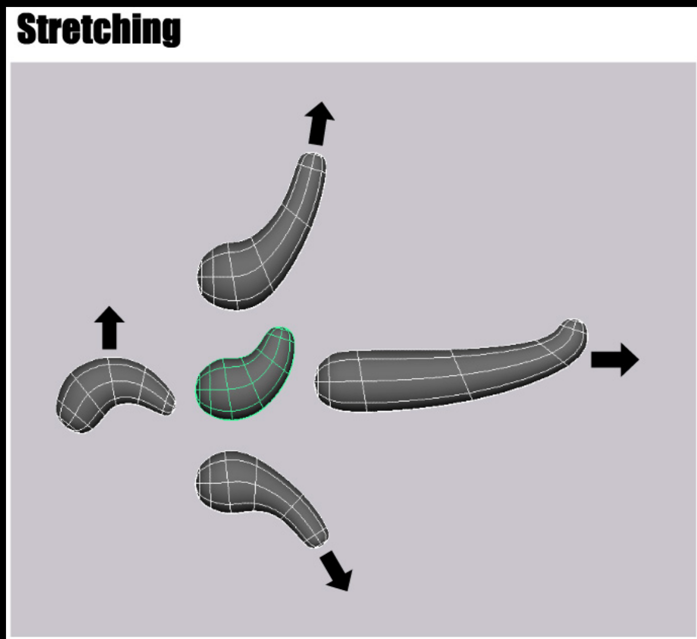


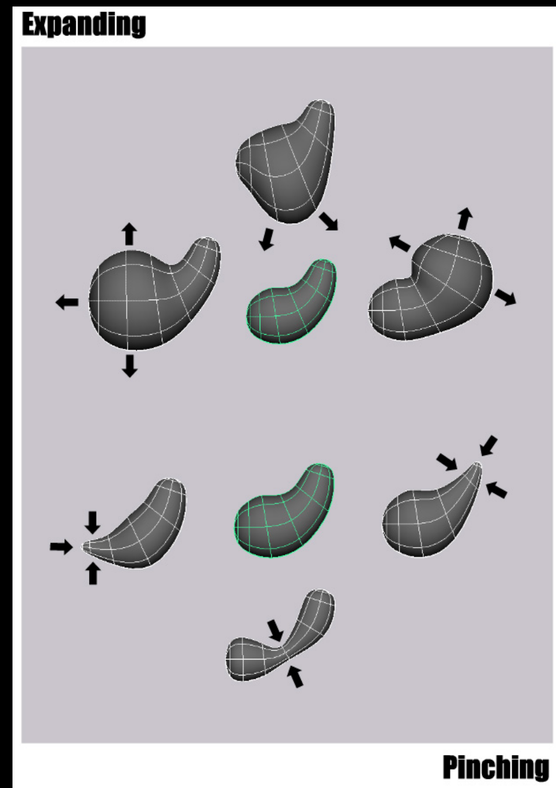


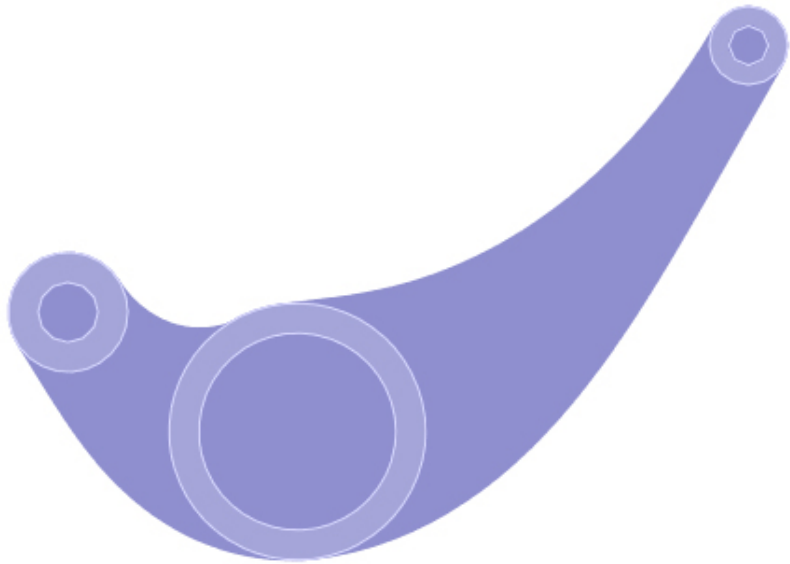


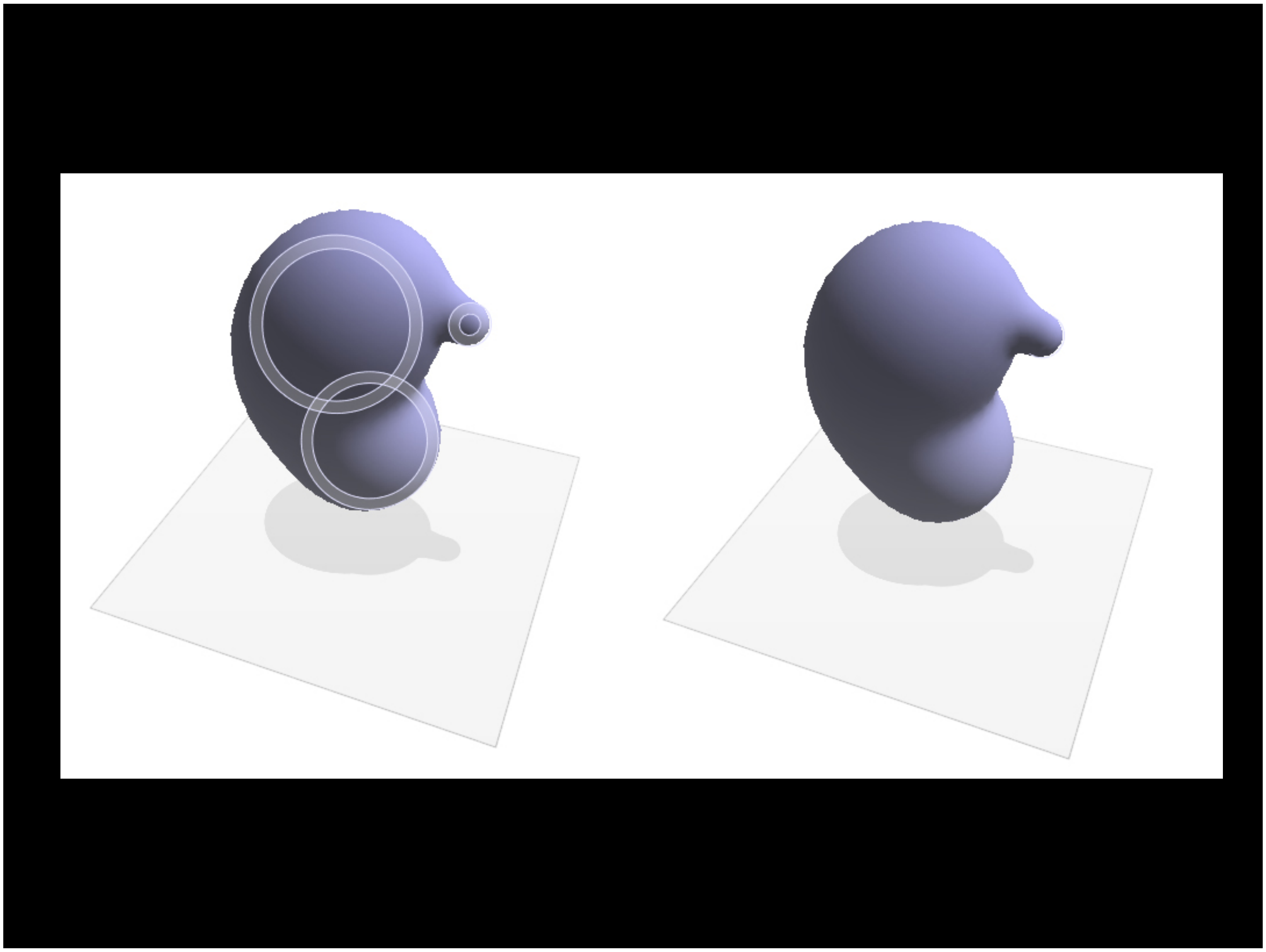


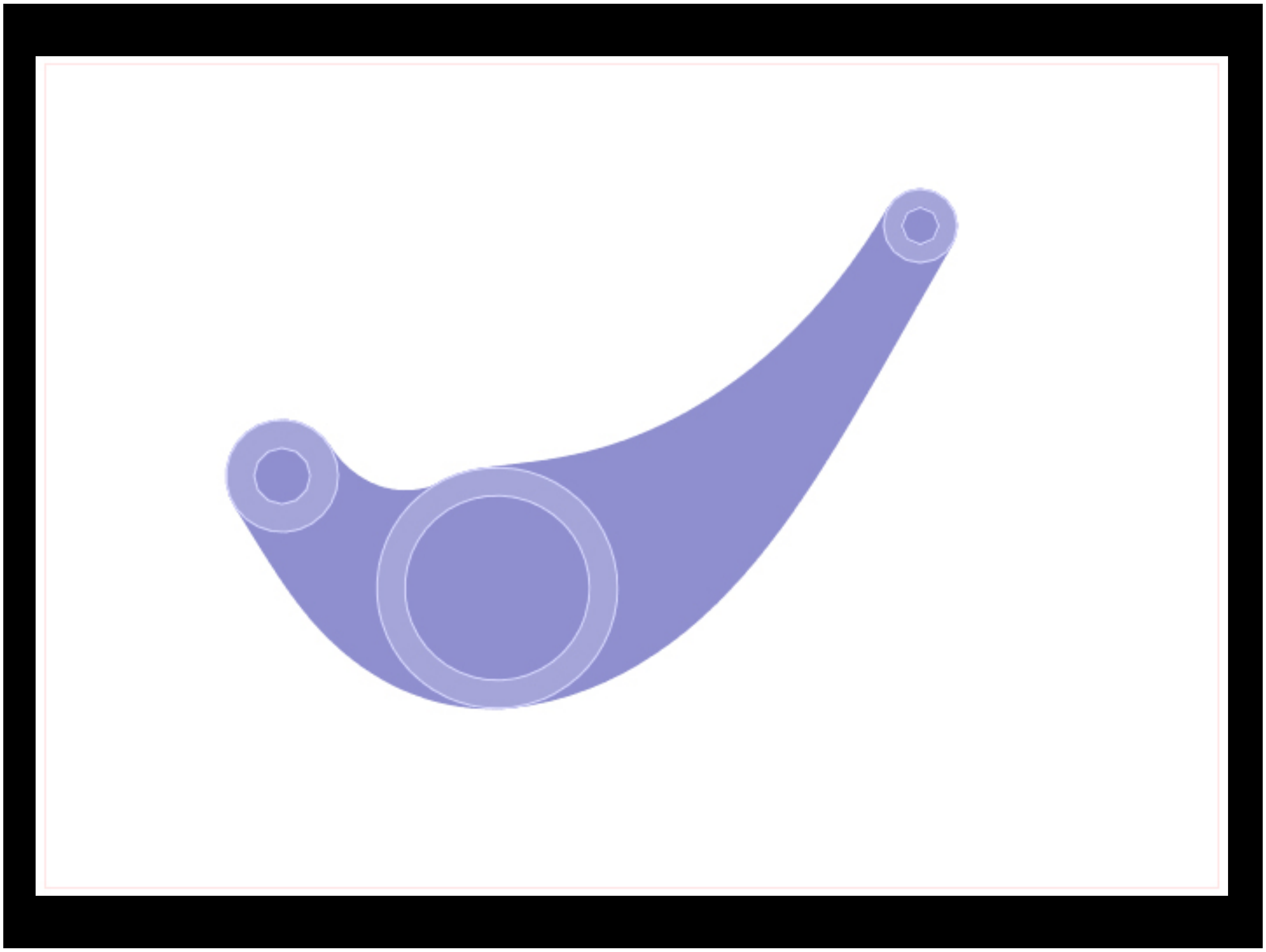


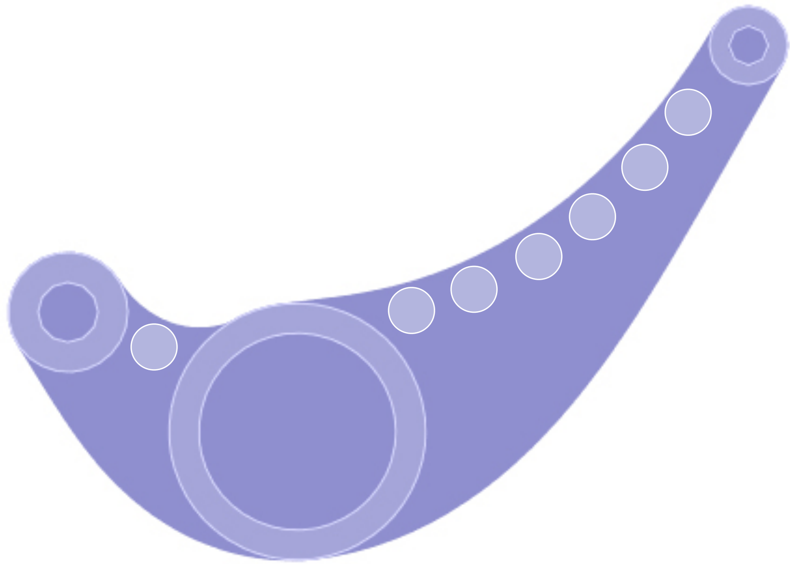

Stretching

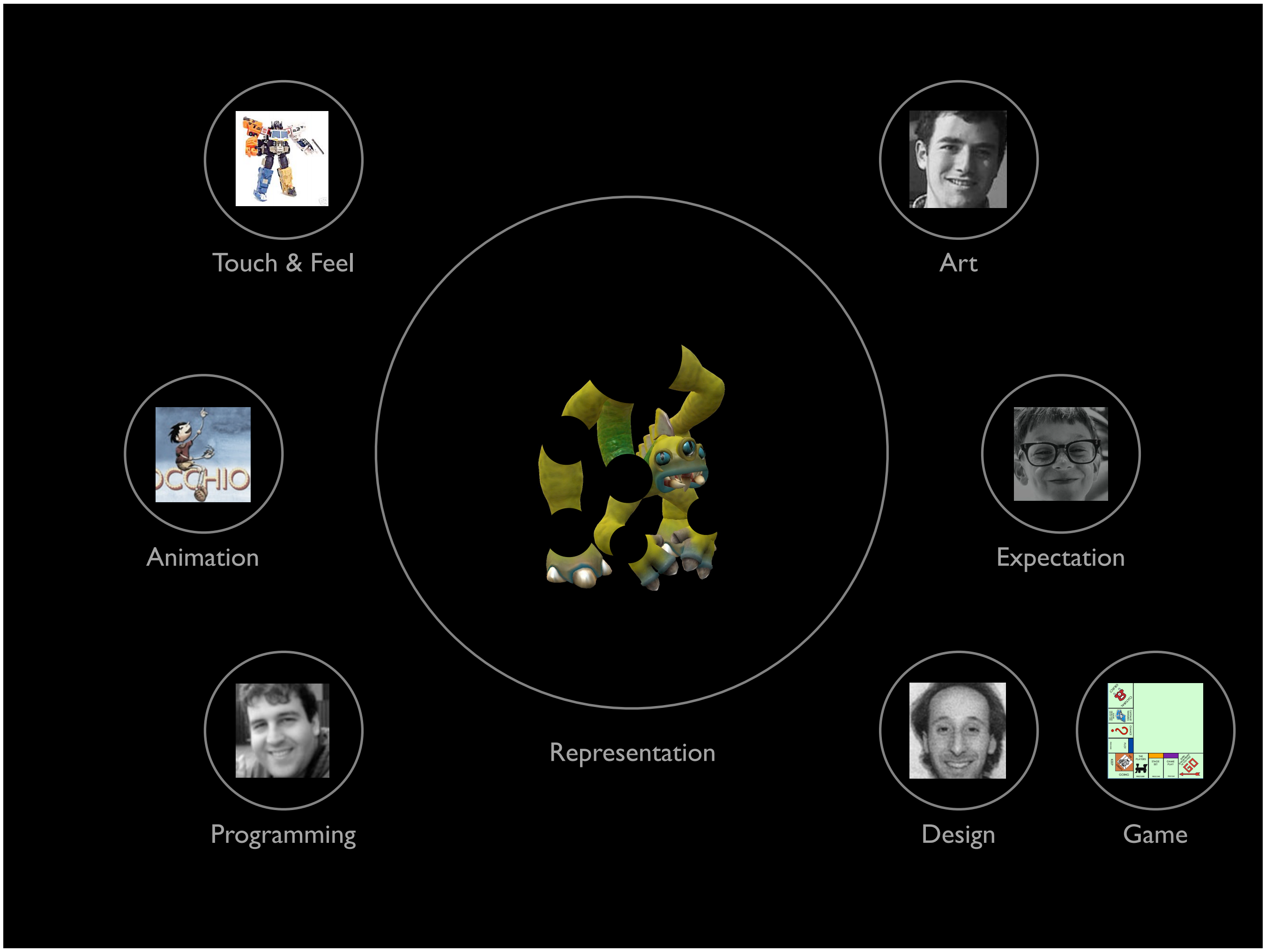

Expanding
Pinching










Touch & Feel
Art
Animation
Expectation
Representation
Programming
Design
Game

So, to wrap up, and tie off this discussion about deep structure, I want to connect this notion of Deep Structure to some other ideas:

• Computer scientists and programmers know that the data structure—the symbolic language you are designing or using—is more important than the algorithm, as it dictates how the algorithm will work.

• And in the contemporary interface design literature: When designing things for humans we must closely track the user's mental model—the user's conceptualization of what is going on, and carefully design and manage that.

• Brian Sutton-Smith, one of the great modern scholars of play, argues that toys—which is what we are dealing with here—have an abstraction and clarity about them that set them off from the background world, and also clearly reveal their inner workings. These schematized forms enable people to easily incorporate them into play, and then repurpose them for new play uses. In fact, this is what happens with Spore's tools:—people use them to make what we expect, this is what draws them in, and then continue to make and do really bizarre and amazing things we did not anticipate.

And to bring us back full circle:

• to Engelbart: PCS, Spore, and Simcity are externalized, automatized, symbol manipulating systems that mesh with the user's symbolic and cognitive processes—with toy like abstractions and other design interventions, to amplify users' ability to do creative and playful tasks—designing aliens, cities, or pinball machines. Players are freed to focus their energies at a high level creative.

• And looking back to the Xerox Alto, again: the importance of articulating user's conceptual model first, and building system from there; not slapping on a user interface afterwards.

[Engelbart process intervention, creative freeing to another process, abstraction: Berlin work, hobbies!, instant cake mix]

Fin

Chaim Gingold
cg@levitylab.com
http://levityab.com