

Unit 4: Conditionals and Artificial Intelligence

Learning Objectives

- 1: The student can use computing tools and techniques to create artifacts
- 3: The student can use computing tools and techniques for creative expression.
- 4: The student can use programming as a creative tool.
- 16: The student can express an algorithm in a language.
- 17: The student can appropriately connect problems and potential algorithmic solutions.
- 19: The student can explain how programs implement algorithms.
- 22: The student can develop a correct program.
- 23: The student can employ appropriate mathematical and logical concepts in programming.
- 30: The student can analyze the beneficial and harmful effects of computing.
- 31: The student can connect computing within economic, social, and cultural contexts.

Readings/Lectures

- Lecture Slides 4.01: Functions ([/bjc-course/curriculum/04-conditionals-and-ai/readings/01-conditional-blocks-slides.pdf](#))
- Reading 4.02: How many days are in a month? (Example) ([/bjc-course/curriculum/04-conditionals-and-ai/readings/02-days-in-month-example](#))
- Lecture Slides 4.03: AI ([/bjc-course/curriculum/04-conditionals-and-ai/readings/03-ai-slides.pdf](#))
- Lecture Video 4.04: If Blocks ([/bjc-course/curriculum/04-conditionals-and-ai/readings/04-if-blocks-video.mp4](#))

External Links

- Cleverbot (<http://www.cleverbot.com/>)
- What is Watson? (http://www.nytimes.com/2010/06/20/magazine/20Computer-t.html?_r=3&) - *New York Times*
- Watson on Jeopardy (<https://www.youtube.com/watch?v=seNkjYyG3gl>) - Youtube
- Computers Solve Checkers (<https://www.scientificamerican.com/article.cfm?id=computers-solve-checkers-its-a-draw>) - *Scientific American*
- Who is ASIMO? (<http://asimo.honda.com/asimotv/>)
- The Google Car (<http://www.forbes.com/sites/chunkamui/2013/01/22/fasten-your-seatbelts-googles-driverless-car-is-worth-trillions/>) - *Forbes*

Labs/Exercises

- Lab 4.01: Random Numbers and Conditionals: Guessing Game ([/bjc-course/curriculum/04-conditionals-and-ai/labs/01-random-numbers-and-conditionals](#))

Conditional Statements

CS Principles

THE IF AND IF...ELSE STRUCTURES

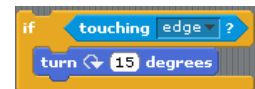
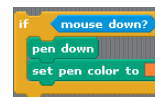
If and If...Else Structures

- Also known as a **Conditional Statement**
- Allows a sprite to make a decision
- Can use sensing tiles to test conditions
- Predicate fields are used for conditions that produce either true or false



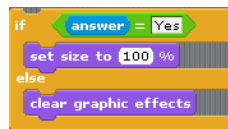
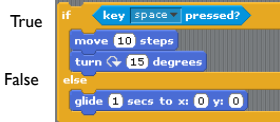
When to use the If Structure

- When the sprite has to make a decision between performing a block of code or not – Yes/No, True/False



When to use the If...Else Structure

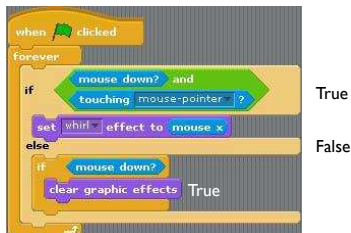
- When the sprite has to make a decision between either performing one block of code or performing a different block of code



NESTED IF'S AND COMPOUND CONDITIONS

Nested If Statements

- When one If-structure is used in another IF-structure, this new structure is known as a **Nested IF**.
- The Nester IF checks two or more conditions.



Logical Operators

- Allow us to put two or more conditions together to create one compound condition
- And, Or and Not
- Order of Precedence: Not, And, Or

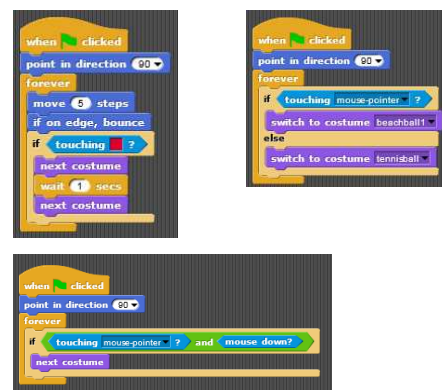


Conditional Operator Example

- All three can be used in any combination.
- Be careful – Don't get confusing. Try to keep them as simple as you can.

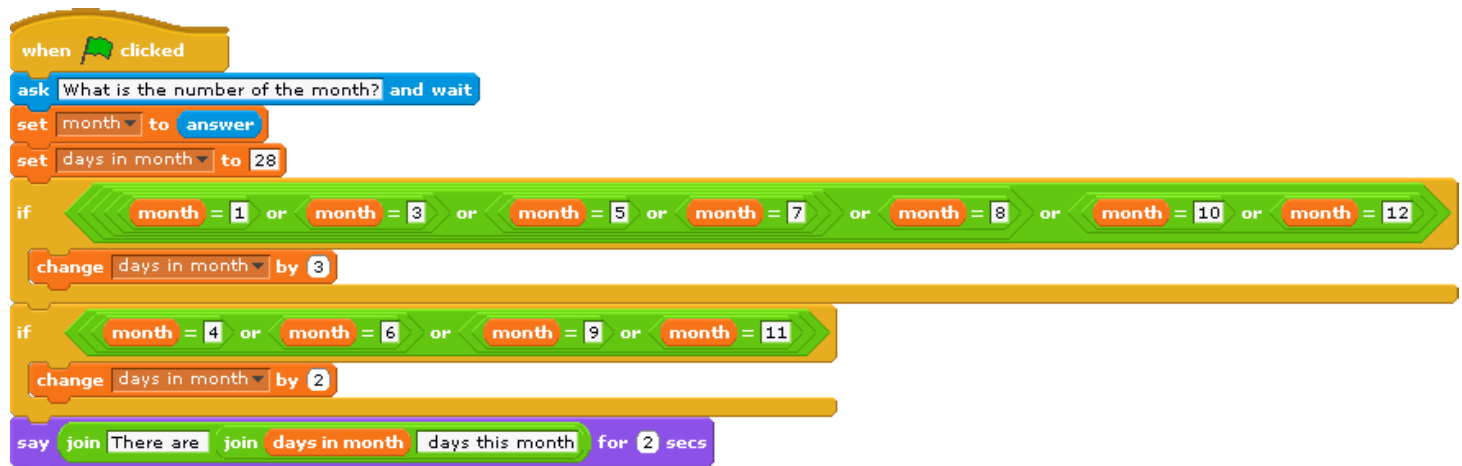


Conditional Examples



How many days are there in a given month?

We have created a Scratch file (`./code/DaysInMonth.sb`) that calculates the number of days in a month based on a month number(eg. 1-12) that a user types in. Discuss with your partner how this Scratch file works. There are a lot of new things in this program that you may not have seen before, so be sure to play around with them.



Note: This code is not written with the best style. We hope your reaction to this is YUCK! We have made the code explicitly complex to help you review some of the blocks from the first lab. In future lab sessions, we will learn to write this type of script more elegantly.

ARTIFICIAL INTELLIGENCE



Artificial Intelligence

- ◉ What is Artificial Intelligence?
 - The Study and Design of Intelligent Agents
- ◉ What is an Intelligent Agent?
 - A system that **perceives** its environment and **takes actions** that maximize its chances of success



Artificial Intelligence

- ◉ What is the **environment** of the Tic Tac Toe paper?
- ◉ What **actions** did it take to maximize its chances of success?

Artificial Intelligence Philosophies

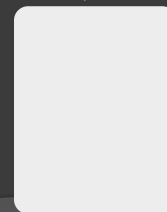
- ◉ Turing
 - If a machine acts as intelligently as a human being, then it is as intelligent as a human being.
 - Judge intelligence of a machine based on its behavior.

Artificial Intelligence Philosophies

- ◉ Searle
 - "The appropriately programmed computer with the right inputs and outputs would thereby have a mind in exactly the same sense human beings have minds."
 - Strong AI – a correct simulation of a mind, IS a mind
 - Machines can act intelligently, and understand in the same sense people do
 - Weak AI – a correct simulation is a model of the mind
 - Machines can only act intelligently, and thus only mimic human behavior

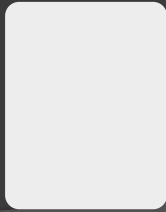
Gender Guessing Game

- ◉ Example) Determine the gender of someone in another room by conversing with them.
 - Can't ask direct answer questions (i.e. are you male/female, etc.)



Turing Test

- Measure of a computer's ability to demonstrate intelligence.
- Consider the Gender Game again...



Turing Test

- If the observer cannot reliably tell the machine from a human, the machine is said to be “intelligent”.
- If a machine is intelligent, it can “think”

Turing Test

- Ultimately says...
- Can computers mimic human behavior?
- This is the goal of AI

Chinese Room Test



- You write a statement in Chinese characters on a piece of paper and slide it under the door.
- You receive a response back to your statement in Chinese.
- Continue until bored.

Chinese Room Test



If you can carry on an intelligent conversation using a piece of paper slid under a door, does this imply that something inside the room understands what you are saying?

Chinese Room Test

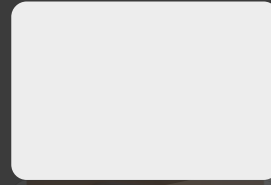
- Who understands Chinese?
- What is the input in this test?
- What is the environment?

Strong AI vs. Weak AI

- ◉ Strong AI
 - The machine **actually** understands Chinese
- ◉ Weak AI
 - The machine **simulates** understanding Chinese

Examples

- ◉ Plen
 - http://www.youtube.com/watch?v=C_UVklCbThI
 - <http://www.youtube.com/watch?v=quzwHqigrDw&NR=1>
 - <http://www.youtube.com/watch?v=dnBbev4-VcE&feature=related>



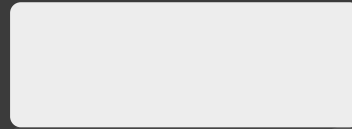
Examples

- ◉ Asimo
 - <http://www.youtube.com/watch?v=NZnqYDDdIW4>
 - <http://www.youtube.com/watch?v=PMw023nS48&feature=related>
 - <http://www.youtube.com/watch?v=0tRo6a4VhvU>



Examples

- ◉ Google Cars
 - <http://www.youtube.com/watch?v=X0i5DHOETFE>



Non-Robotics Examples

- ◉ Checkers
- ◉ Deep Blue
- ◉ Watson
 - <http://www.youtube.com/watch?v=ZLdkJpAtt1I>
 - 2:00

This copy is for your personal, noncommercial use only. You can order presentation-ready copies for distribution to your colleagues, clients or customers [here](#) or use the "Reprints" tool that appears next to any article. Visit www.nytreprints.com for samples and additional information. [Order a reprint of this article now.](#)

June 16, 2010

What Is I.B.M.'s Watson?

By CLIVE THOMPSON

“Toured the Burj in this U.A.E. city. They say it’s the tallest tower in the world; looked over the ledge and lost my lunch.”

This is the quintessential sort of clue you hear on the TV game show “Jeopardy!” It’s witty (the clue’s category is “Postcards From the Edge”), demands a large store of trivia and requires contestants to make confident, split-second decisions. This particular clue appeared in a mock version of the game in December, held in Hawthorne, N.Y. at one of I.B.M.’s research labs. Two contestants — Dorothy Gilmartin, a health teacher with her hair tied back in a ponytail, and Alison Kolani, a copy editor — furrowed their brows in concentration. Who would be the first to answer?

Neither, as it turned out. Both were beaten to the buzzer by the third combatant: Watson, a supercomputer.

For the last three years, I.B.M. scientists have been developing what they expect will be the world’s most advanced “[question answering](#)” machine, able to understand a question posed in everyday human elocution — “natural language,” as computer scientists call it — and respond with a precise, factual answer. In other words, it must do more than what search engines like [Google](#) and Bing do, which is merely point to a document where you might find the answer. It has to pluck out the correct answer itself. Technologists have long regarded this sort of artificial intelligence as a holy grail, because it would allow machines to converse more naturally with people, letting us ask questions instead of typing keywords. Software firms and university scientists have produced question-answering systems for years, but these have mostly been limited to simply phrased questions. Nobody ever tackled “Jeopardy!” because experts assumed that even for the latest artificial intelligence, the game was simply too hard: the clues are too puzzling and allusive, and the breadth of trivia is too wide.

With Watson, I.B.M. claims it has cracked the problem — and aims to prove as much on national TV. The producers of “Jeopardy!” have agreed to pit Watson against some of the game’s best former players as early as this fall. To test Watson’s capabilities against actual humans, I.B.M.’s scientists began holding live matches last winter. They mocked up a conference room to resemble the actual “Jeopardy!” set, including buzzers and stations for

the human contestants, brought in former contestants from the show and even hired a host for the occasion: Todd Alan Crain, who plays a newscaster on the satirical Onion News Network.

Technically speaking, Watson wasn't in the room. It was one floor up and consisted of a roomful of servers working at speeds thousands of times faster than most ordinary desktops. Over its three-year life, Watson stored the content of tens of millions of documents, which it now accessed to answer questions about almost anything. (Watson is not connected to the Internet; like all "Jeopardy!" competitors, it knows only what is already in its "brain.") During the sparring matches, Watson received the questions as electronic texts at the same moment they were made visible to the human players; to answer a question, Watson spoke in a machine-synthesized voice through a small black speaker on the game-show set. When it answered the Burj clue — "What is Dubai?" ("Jeopardy!" answers must be phrased as questions) — it sounded like a perkier cousin of the computer in the movie "WarGames" that nearly destroyed the world by trying to start a nuclear war.

This time, though, the computer was doing the right thing. Watson won \$1,000 (in pretend money, anyway), pulled ahead and eventually defeated Gilmartin and Kolani soundly, winning \$18,400 to their \$12,000 each.

"Watson," Crain shouted, "is our new champion!"

It was just the beginning. Over the rest of the day, Watson went on a tear, winning four of six games. It displayed remarkable facility with cultural trivia ("This action flick starring Roy Scheider in a high-tech police helicopter was also briefly a TV series" — "What is 'Blue Thunder'?"), science ("The greyhound originated more than 5,000 years ago in this African country, where it was used to hunt gazelles" — "What is Egypt?") and sophisticated wordplay ("Classic candy bar that's a female [Supreme Court](#) justice" — "What is Baby Ruth Ginsburg?").

By the end of the day, the seven human contestants were impressed, and even slightly unnerved, by Watson. Several made references to Skynet, the computer system in the "Terminator" movies that achieves consciousness and decides humanity should be destroyed. "My husband and I talked about what my role in this was," Samantha Boardman, a graduate student, told me jokingly. "Was I the thing that was going to help the A.I. become aware of itself?" She had distinguished herself with her swift responses to the "Rhyme Time" puzzles in one of her games, winning nearly all of them before Watson could figure out the clues, but it didn't help. The computer still beat her three times. In one game, she finished with no money.

“He plays to win,” Boardman said, shaking her head. “He’s really not messing around!” Like most of the contestants, she had started calling Watson “he.”

WE LIVE IN AN AGE of increasingly smart machines. In recent years, engineers have pushed into areas, from voice recognition to robotics to search engines, that once seemed to be the preserve of humans. But I.B.M. has a particular knack for pitting man against machine. In 1997, the company’s supercomputer Deep Blue famously beat the grandmaster [Garry Kasparov](#) at chess, a feat that generated enormous publicity for I.B.M. It did not, however, produce a marketable product; the technical accomplishment — playing chess really well — didn’t translate to real-world business problems and so produced little direct profit for I.B.M. In the mid ’00s, the company’s top executives were looking for another high-profile project that would provide a similar flood of global publicity. But this time, they wanted a “grand challenge” (as they call it internally), that would meet a real-world need.

Question-answering seemed to be a good fit. In the last decade, question-answering systems have become increasingly important for firms dealing with mountains of documents. Legal firms, for example, need to quickly sift through case law to find a useful precedent or citation; help-desk workers often have to negotiate enormous databases of product information to find an answer for an agitated customer on the line. In situations like these, speed can often be of the essence; in the case of help desks, labor is billed by the minute, so high-tech firms with slender margins often lose their profits providing telephone support. How could I.B.M. push question-answering technology further?

When one I.B.M. executive suggested taking on “Jeopardy!” he was immediately pooh-poohed. Deep Blue was able to play chess well because the game is perfectly logical, with fairly simple rules; it can be reduced easily to math, which computers handle superbly. But the rules of language are much trickier. At the time, the very best question-answering systems — some created by software firms, some by university researchers — could sort through news articles on their own and answer questions about the content, but they understood only questions stated in very simple language (“What is the capital of Russia?”); in government-run competitions, the top systems answered correctly only about 70 percent of the time, and many were far worse. “Jeopardy!” with its witty, punning questions, seemed beyond their capabilities. What’s more, winning on “Jeopardy!” requires finding an answer in a few seconds. The top question-answering machines often spent longer, even entire minutes, doing the same thing.

“The reaction was basically, ‘No, it’s too hard, forget it, no way can you do it,’ ” David Ferrucci told me not long ago. Ferrucci, I.B.M.’s senior manager for its Semantic Analysis and Integration department, heads the Watson project, and I met him for the first time last November at I.B.M.’s lab. An artificial-intelligence researcher who has long specialized in

question-answering systems, Ferrucci chafed at the slow progress in the field. A fixture in the office in the evenings and on weekends, he is witty, voluble and intense. While dining out recently, his wife asked the waiter if Ferrucci's meal included any dairy. "Is he lactose intolerant?" the waiter inquired. "Yes," his wife replied, "and just generally intolerable." Ferrucci told me he was recently prescribed a mouth guard because the stress of watching Watson play had him clenching his teeth excessively.

Ferrucci was never an aficionado of "Jeopardy!" ("I've certainly seen it," he said with a shrug. "I'm not a big fan.") But he craved an ambitious goal that would impel him to break new ground, that would verge on science fiction, and this fit the bill. "The computer on 'Star Trek' is a question-answering machine," he says. "It understands what you're asking and provides just the right chunk of response that you needed. When is the computer going to get to a point where the computer knows how to talk to you? That's my question."

What makes language so hard for computers, Ferrucci explained, is that it's full of "intended meaning." When people decode what someone else is saying, we can easily unpack the many nuanced allusions and connotations in every sentence. He gave me an example in the form of a "Jeopardy!" clue: "The name of this hat is elementary, my dear contestant." People readily detect the wordplay here — the echo of "elementary, my dear Watson," the famous phrase associated with [Sherlock Holmes](#) — and immediately recall that the Hollywood version of Holmes sports a deerstalker hat. But for a computer, there is no simple way to identify "elementary, my dear contestant" as wordplay. Cleverly matching different keywords, and even different fragments of the sentence — which in part is how most search engines work these days — isn't enough, either. (Type that clue into Google, and you'll get first-page referrals to "elementary, my dear watson" but none to deerstalker hats.)

What's more, even if a computer determines that the actual underlying question is "What sort of hat does Sherlock Holmes wear?" its data may not be stored in such a way that enables it to extract a precise answer. For years, computer scientists built question-answering systems by creating specialized databases, in which certain facts about the world were recorded and linked together. You could do this with Sherlock Holmes by building a database that includes connections between catchphrases and his hat and his violin-playing. But that database would be pretty narrow; it wouldn't be able to answer questions about nuclear power, or fish species, or the history of France. Those would require their own hand-made databases. Pretty soon you'd face the impossible task of organizing all the information known to man — of "boiling the ocean," as Ferrucci put it. In computer science, this is known as a "bottleneck" problem. And even if you could get past it, you might then face the issue of "brittleness": if your database contains only facts you input manually, it breaks any time you ask it a question about something beyond that

material. There's no way to hand-write a database that would include the answer to every "Jeopardy!" clue, because the subject matter is potentially all human knowledge.

The great shift in artificial intelligence began in the last 10 years, when computer scientists began using statistics to analyze huge piles of documents, like books and news stories. They wrote algorithms that could take any subject and automatically learn what types of words are, statistically speaking, most (and least) associated with it. Using this method, you could put hundreds of articles and books and movie reviews discussing Sherlock Holmes into the computer, and it would calculate that the words "deerstalker hat" and "Professor Moriarty" and "opium" are frequently correlated with one another, but not with, say, the [Super Bowl](#). So at that point you could present the computer with a question that didn't mention Sherlock Holmes by name, but if the machine detected certain associated words, it could conclude that Holmes was the probable subject — and it could also identify hundreds of other concepts and words that weren't present but that were likely to be related to Holmes, like "Baker Street" and "chemistry."

In theory, this sort of statistical computation has been possible for decades, but it was impractical. Computers weren't fast enough, memory wasn't expansive enough and in any case there was no easy way to put millions of documents into a computer. All that changed in the early '00s. Computer power became drastically cheaper, and the amount of online text exploded as millions of people wrote blogs and wikis about anything and everything; news organizations and academic journals also began putting all their works in digital format. What's more, question-answering experts spent the previous couple of decades creating several linguistic tools that helped computers puzzle through language — like rhyming dictionaries, bulky synonym finders and "classifiers" that recognized the parts of speech.

Still, the era's best question-answering systems remained nowhere near being able to take on "Jeopardy!" In 2006, Ferrucci tested I.B.M.'s most advanced system — it wasn't the best in its field but near the top — by giving it 500 questions from previous shows. The results were dismal. He showed me a chart, prepared by I.B.M., of how real-life "Jeopardy!" champions perform on the TV show. They are clustered at the top in what Ferrucci calls "the winner's cloud," which consists of individuals who are the first to hit the buzzer about 50 percent of the time and, after having "won" the buzz, solve on average 85 to 95 percent of the clues. In contrast, the I.B.M. system languished at the bottom of the chart. It was rarely confident enough to answer a question, and when it was, it got the right answer only 15 percent of the time. Humans were fast and smart; I.B.M.'s machine was slow and dumb.

"Humans are just — boom! — they're just plowing through this in just seconds," Ferrucci said excitedly. "They're getting the questions, they're breaking them down, they're interpreting them, they're getting the right interpretation, they're looking this up in their

memory, they're scoring, they're doing all this just instantly."

But Ferrucci argued that I.B.M. could be the one to finally play "Jeopardy!" If the firm focused its computer firepower — including its new "BlueGene" servers — on the challenge, Ferrucci could conduct experiments dozens of times faster than anyone had before, allowing him to feed more information into Watson and test new algorithms more quickly. Ferrucci was ambitious for personal reasons too: if he didn't try this, another computer scientist might — "and then bang, you are irrelevant," he told me.

"I had no interest spending the next five years of my life pursuing things in the small," he said. "I wanted to push the limits." If they could succeed at "Jeopardy!" soon after that they could bring the underlying technology to market as customizable question-answering systems. In 2007, his bosses gave him three to five years and increased his team to 15 people.

FERRUCCI'S MAIN breakthrough was not the design of any single, brilliant new technique for analyzing language. Indeed, many of the statistical techniques Watson employs were already well known by computer scientists. One important thing that makes Watson so different is its enormous speed and memory. Taking advantage of I.B.M.'s supercomputing heft, Ferrucci's team input millions of documents into Watson to build up its knowledge base — including, he says, "books, reference material, any sort of dictionary, thesauri, folksonomies, taxonomies, encyclopedias, any kind of reference material you can imagine getting your hands on or licensing. Novels, bibles, plays."

Watson's speed allows it to try thousands of ways of simultaneously tackling a "Jeopardy!" clue. Most question-answering systems rely on a handful of algorithms, but Ferrucci decided this was why those systems do not work very well: no single algorithm can simulate the human ability to parse language and facts. Instead, Watson uses more than a hundred algorithms at the same time to analyze a question in different ways, generating hundreds of possible solutions. Another set of algorithms ranks these answers according to plausibility; for example, if dozens of algorithms working in different directions all arrive at the same answer, it's more likely to be the right one. In essence, Watson thinks in probabilities. It produces not one single "right" answer, but an enormous number of possibilities, then ranks them by assessing how likely each one is to answer the question.

Ferrucci showed me how Watson handled this sample "Jeopardy!" clue: "He was presidentially pardoned on Sept. 8, 1974." In the first pass, the algorithms came up with "Nixon." To evaluate whether "Nixon" was the best response, Watson performed a clever trick: it inserted the answer into the original phrase — "Nixon was presidentially pardoned on Sept. 8, 1974" — and then ran it as a new search, to see if it also produced results that supported "Nixon" as the right answer. (It did. The new search returned the result "Ford

pardoned Nixon on Sept. 8, 1974,” a phrasing so similar to the original clue that it helped make “Nixon” the top-ranked solution.)

Other times, Watson uses algorithms that can perform basic cross-checks against time or space to help detect which answer seems better. When the computer analyzed the clue “In 1594 he took a job as a tax collector in Andalusia,” the two most likely answers generated were “Thoreau” and “Cervantes.” Watson assessed “Thoreau” and discovered his birth year was 1817, at which point the computer ruled him out, because he wasn’t alive in 1594. “Cervantes” became the top-ranked choice.

When Watson is playing a game, Ferrucci lets the audience peek into the computer’s analysis. A monitor shows Watson’s top five answers to a question, with a bar graph beside each indicating its confidence. During one of my visits, the host read the clue “Thousands of prisoners in the Philippines re-enacted the moves of the video of this [Michael Jackson](#) hit.” On the monitor, I could see that Watson’s top pick was “Thriller,” with a confidence level of roughly 80 percent. This answer was correct, and Watson buzzed first, so it won \$800. Watson’s next four choices — “Music video,” “Billie Jean,” “Smooth Criminal” and “MTV” — had only slivers for their bar graphs. It was a fascinating glimpse into the machine’s workings, because you could spy the connective thread running between the possibilities, even the wrong ones. “Billie Jean” and “Smooth Criminal” were also major hits by Michael Jackson, and “MTV” was the main venue for his videos. But it’s very likely that none of those correlated well with “Philippines.”

After a year, Watson’s performance had moved halfway up to the “winner’s cloud.” By 2008, it had edged into the cloud; on paper, anyway, it could beat some of the lesser “Jeopardy!” champions. Confident they could actually compete on TV, I.B.M. executives called up Harry Friedman, the executive producer of “Jeopardy!” and raised the possibility of putting Watson on the air.

Friedman told me he and his fellow executives were surprised: nobody had ever suggested anything like this. But they quickly accepted the challenge. “Because it’s I.B.M., we took it seriously,” Friedman said. “They had the experience with Deep Blue and the chess match that became legendary.”

WHEN THEY FIRST showed up to play Watson, many of the contestants worried that they didn’t stand a chance. Human memory is frail. In a high-stakes game like “Jeopardy!” players can panic, becoming unable to recall facts they would otherwise remember without difficulty. Watson doesn’t have this problem. It might have trouble with its analysis or be unable to logically connect a relevant piece of text to a question. But it doesn’t forget things. Plus, it has lightning-fast reactions — wouldn’t it simply beat the humans to the buzzer every time?

“We’re relying on nerves — old nerves,” Dorothy Gilmartin complained, halfway through her first game, when it seemed that Watson was winning almost every buzz.

Yet the truth is, in more than 20 games I witnessed between Watson and former “Jeopardy!” players, humans frequently beat Watson to the buzzer. Their advantage lay in the way the game is set up. On “Jeopardy!” when a new clue is given, it pops up on screen visible to all. (Watson gets the text electronically at the same moment.) But contestants are not allowed to hit the buzzer until the host is finished reading the question aloud; on average, it takes the host about six or seven seconds to read the clue.

Players use this precious interval to figure out whether or not they have enough confidence in their answers to hazard hitting the buzzer. After all, buzzing carries a risk: someone who wins the buzz on a \$1,000 question but answers it incorrectly loses \$1,000.

Often those six or seven seconds weren’t enough time for Watson. The humans reacted more quickly. For example, in one game an \$800 clue was “In Poland, pick up some *kalafior* if you crave this broccoli relative.” A human contestant jumped on the buzzer as soon as he could. Watson, meanwhile, was still processing. Its top five answers hadn’t appeared on the screen yet. When these finally came up, I could see why it took so long. Something about the question had confused the computer, and its answers came with mere slivers of confidence. The top two were “vegetable” and “cabbage”; the correct answer — “cauliflower” — was the third guess.

To avoid losing money — Watson doesn’t care about the money, obviously; winnings are simply a way for I.B.M. to see how fast and accurately its system is performing — Ferrucci’s team has programmed Watson generally not to buzz until it arrives at an answer with a high confidence level. In this regard, Watson is actually at a disadvantage, because the best “Jeopardy!” players regularly hit the buzzer as soon as it’s possible to do so, even if it’s before they’ve figured out the clue. “Jeopardy!” rules give them five seconds to answer after winning the buzz. So long as they have a good feeling in their gut, they’ll pounce on the buzzer, trusting that in those few extra seconds the answer will pop into their heads. Ferrucci told me that the best human contestants he had brought in to play against Watson were amazingly fast. “They can buzz in 10 milliseconds,” he said, sounding astonished. “Zero milliseconds!”

On the third day I watched Watson play, it did quite poorly, losing four of seven games, in one case without any winnings at all. Often Watson appeared to misunderstand the clue and offered answers so inexplicable that the audience erupted in laughter. Faced with the clue “This ‘insect’ of a gangster was a real-life hit man for Murder Incorporated in the 1930s & ’40s,” Watson responded with “James Cagney.” Up on the screen, I could see that none of its lesser choices were the correct one, “Bugsy Siegel.” Later, when asked to complete the

phrase “Toto, I’ve a feeling we’re not in Ka—,” Watson offered “not in Kansas anymore,” which was incorrect, since the precise phrasing was simply “Kansas anymore,” and “Jeopardy!” is strict about phrasings. When I looked at the screen, I noticed that the answers Watson had ranked lower were pretty odd, including “Steve Porcaro,” the keyboardist for the band Toto (which made a vague sort of sense), and “[Jackie Chan](#)” (which really didn’t). In another game, Watson’s logic appeared to fall down some odd semantic rabbit hole, repeatedly giving the answer “[Tommy Lee Jones](#)” — the name of the Hollywood actor — to several clues that had nothing to do with him.

In the corner of the conference room, Ferrucci sat typing into a laptop. Whenever Watson got a question wrong, Ferrucci winced and stamped his feet in frustration, like a college-football coach watching dropped passes. “This is *torture*,” he added, laughing.

Seeing Watson’s errors, you can sometimes get a sense of its cognitive shortcomings. For example, in “Jeopardy!” the category heading often includes a bit of wordplay that explains how the clues are to be addressed. Watson sometimes appeared to mistakenly analyze the entire category and thus botch every clue in it. One game included the category “Stately Botanical Gardens,” which indicated that every clue would list several gardens, and the answer was the relevant state. Watson clearly didn’t grasp this; it answered “botanic garden” repeatedly. I also noticed that when Watson was faced with very short clues — ones with only a word or two — it often seemed to lose the race to the buzzer, possibly because the host read the clues so quickly that Watson didn’t have enough time to do its full calculations. The humans, in contrast, simply trusted their guts and jumped.

Ferrucci refused to talk on the record about Watson’s blind spots. He’s aware of them; indeed, his team does “error analysis” after each game, tracing how and why Watson messed up. But he is terrified that if competitors knew what types of questions Watson was bad at, they could prepare by boning up in specific areas. I.B.M. required all its sparring-match contestants to sign nondisclosure agreements prohibiting them from discussing their own observations on what, precisely, Watson was good and bad at. I signed no such agreement, so I was free to describe what I saw; but Ferrucci wasn’t about to make it easier for me by cataloguing Watson’s vulnerabilities.

Computer scientists I spoke to agreed that witty, allusive clues will probably be Watson’s weak point. “Retrieval of obscure Italian poets is easy — [Watson] will never forget that one,” Peter Norvig, the director of research at Google, told me. “But ‘Jeopardy!’ tends to have a lot of wordplay, and that’s going to be a challenge.” Certainly on many occasions this seemed to be true. Still, at other times I was startled by Watson’s eerily humanlike ability to untangle astonishingly coy clues. During one game, a category was “All-Eddie Before & After,” indicating that the clue would hint at two different things that need to be blended together, one of which included the name “Eddie.” The \$2,000 clue was “A ‘Green Acres’

star goes existential (& French) as the author of 'The Fall.' " Watson nailed it perfectly: "Who is Eddie Albert Camus?"

Ultimately, Watson's greatest edge at "Jeopardy!" probably isn't its perfect memory or lightning speed. It is the computer's lack of emotion. "Managing your emotions is an enormous part of doing well" on "Jeopardy!" Bob Harris, a five-time champion, told me. "Every single time I've ever missed a Daily Double, I always miss the next clue, because I'm still kicking myself." Because there is only a short period before the next clue comes along, the stress can carry over. Similarly, humans can become much more intimidated by a \$2,000 clue than a \$200 one, because the more expensive clues are presumably written to be much harder.

Whether Watson will win when it goes on TV in a real "Jeopardy!" match depends on whom "Jeopardy!" pits against the computer. Watson will not appear as a contestant on the regular show; instead, "Jeopardy!" will hold a special match pitting Watson against one or more famous winners from the past. If the contest includes Ken Jennings — the best player in "Jeopardy!" history, who won 74 games in a row in 2004 — Watson will lose if its performance doesn't improve. It's pretty far up in the winner's cloud, but it's not yet at Jennings's level; in the sparring matches, Watson was beaten several times by opponents who did nowhere near as well as Jennings. (Indeed, it sometimes lost to people who hadn't placed first in their own appearances on the show.) The show's executive producer, Harry Friedman, will not say whom it is picking to play against Watson, but he refused to let Jennings be interviewed for this story, which is suggestive.

Ferrucci says his team will continue to fine-tune Watson, but improving its performance is getting harder. "When we first started, we'd add a new algorithm and it would improve the performance by 10 percent, 15 percent," he says. "Now it'll be like half a percent is a good improvement."

Ferrucci's attitude toward winning is conflicted. I could see that he hungers to win. And losing badly on national TV might mean negative publicity for I.B.M. But Ferrucci also argued that Watson might lose merely because of bad luck. Should one of Watson's opponents land on both Daily Doubles, for example, that player might double his or her money and vault beyond Watson's ability to catch up, even if the computer never flubs another question.

Ultimately, Ferrucci claimed not to worry about winning or losing. He told me he's happy that I.B.M. has simply pushed this far and produced a system that performs so well at answering questions. Even a televised flameout, he said, won't diminish the street cred Watson will give I.B.M. in the computer-science field. "I don't really care about 'Jeopardy!'" he told me, shrugging.

I.B.M. PLANS TObegin selling versions of Watson to companies in the next year or two. John Kelly, the head of I.B.M.'s research labs, says that Watson could help decision-makers sift through enormous piles of written material in seconds. Kelly says that its speed and quality could make it part of rapid-fire decision-making, with users talking to Watson to guide their thinking process.

"I want to create a medical version of this," he adds. "A Watson M.D., if you will." He imagines a hospital feeding Watson every new medical paper in existence, then having it answer questions during split-second emergency-room crises. "The problem right now is the procedures, the new procedures, the new medicines, the new capability is being generated faster than physicians can absorb on the front lines and it can be deployed." He also envisions using Watson to produce virtual call centers, where the computer would talk directly to the customer and generally be the first line of defense, because, "as you've seen, this thing can answer a question faster and more accurately than most human beings."

"I want to create something that I can take into every other retail industry, in the transportation industry, you name it, the banking industry," Kelly goes on to say. "Any place where time is critical and you need to get advanced state-of-the-art information to the front of decision-makers. Computers need to go from just being back-office calculating machines to improving the intelligence of people making decisions." At first, a Watson system could cost several million dollars, because it needs to run on at least one \$1 million I.B.M. server. But Kelly predicts that within 10 years an artificial brain like Watson could run on a much cheaper server, affordable by any small firm, and a few years after that, on a laptop.

Ted Senator, a vice president of SAIC — a high-tech firm that frequently helps design government systems — is a former "Jeopardy!" champion and has followed Watson's development closely; in October he visited I.B.M. and played against Watson himself. (He lost.) He says that Watson-level artificial intelligence could make it significantly easier for citizens to get answers quickly from massive, ponderous bureaucracies. He points to the recent "[cash for clunkers](#)" program. He tried to participate, but when he went to the government site to see if his car qualified, he couldn't figure it out: his model, a 1995 Saab 9000, was listed twice, each time with different mileage-per-gallon statistics. What he needed was probably buried deep inside some government database, but the bureaucrats hadn't presented the information clearly enough. "So I gave up," he says. This is precisely the sort of task a Watson-like artificial intelligence can assist in, he says. "You can imagine if I'm applying for health insurance, having to explain the details of my personal situation, or if I'm trying to figure out if I'm eligible for a particular tax deduction. Any place there's massive data that surpasses the human's ability to sort through it, and there's a time constraint on getting an answer."

Many experts imagine even quirkier ways that everyday life might be transformed as

question-answering technology becomes more powerful and widespread. Andrew Hickl, the C.E.O. of Language Computer Corporation, which makes question-answering systems, among other things, for businesses, was recently asked by a client to make a “contradiction engine”: if you tell it a statement, it tries to find evidence on the Web that contradicts it. “It’s like, ‘I believe that Dallas is the most beautiful city in the United States,’ and I want to find all the evidence on the Web that contradicts that.” (It produced results that were only 70 percent relevant, which satisfied his client.) Hickl imagines people using this sort of tool to read through the daily news. “We could take something that [Harry Reid](#) says and immediately figure out what contradicts it. Or somebody tweets something that’s wrong, and we could automatically post a tweet saying, ‘No, actually, that’s wrong, and here’s proof.’ ”

CULTURALLY, OF COURSE, advances like Watson are bound to provoke nervous concerns too. High-tech critics have begun to wonder about the wisdom of relying on artificial-intelligence systems in the face of complex reality. Many Wall Street firms, for example, now rely on “millisecond trading” computers, which detect deviations in prices and order trades far faster than humans ever could; but these are now regarded as a possible culprit in the seemingly irrational hourlong stock-market plunge of the spring. Would doctors in an E.R. feel comfortable taking action based on a split-second factual answer from a Watson M.D.? And while service companies can clearly save money by relying more on question-answering systems, they are precisely the sort of labor-saving advance deplored by unions — and customers who crave the ability to talk to a real, intelligent human on the phone.

Some scientists, moreover, argue that Watson has serious limitations that could hamper its ability to grapple with the real world. It can analyze texts and draw basic conclusions from the facts it finds, like figuring out if one event happened later than another. But many questions we want answered require more complex forms of analysis. Last year, the computer scientist Stephen Wolfram released “Wolfram Alpha,” a question-answering engine that can do mathematical calculations about the real world. Ask it to “compare the populations of New York City and Cincinnati,” for example, and it will not only give you their populations — 8.4 million versus 333,336 — it will also create a bar graph comparing them visually and calculate their ratio (25.09 to 1) and the percentage relationship between them (New York is 2,409 percent larger). But this sort of automated calculation is only possible because Wolfram and his team spent years painstakingly hand-crafting databases in a fashion that enables a computer to perform this sort of analysis — by typing in the populations of New York and Cincinnati, for example, and tagging them both as “cities” so that the engine can compare them. This, Wolfram says, is the deep challenge of artificial intelligence: a lot of human knowledge isn’t represented in words alone, and a computer won’t learn that stuff just by encoding English language texts, as Watson does. The only

way to program a computer to do this type of mathematical reasoning might be to do precisely what Ferrucci doesn't want to do — sit down and slowly teach it about the world, one fact at a time.

“Not to take anything away from this ‘Jeopardy!’ thing, but I don’t think Watson really is answering questions — it’s not like the ‘Star Trek’ computer,” Wolfram says. (Of course, Wolfram Alpha cannot answer the sort of broad-ranging trivia questions that Watson can, either, because Wolfram didn’t design it for that purpose.) What’s more, Watson can answer only questions asking for an objectively knowable fact. It cannot produce an answer that requires judgment. It cannot offer a new, unique answer to questions like “What’s the best high-tech company to invest in?” or “When will there be peace in the Middle East?” All it will do is look for source material in its database that appears to have addressed those issues and then collate and compose a string of text that seems to be a statistically likely answer. Neither Watson nor Wolfram Alpha, in other words, comes close to replicating human wisdom.

At best, Ferrucci suspects that Watson might be simulating, in a stripped-down fashion, some of the ways that our human brains process language. Modern neuroscience has found that our brain is highly “parallel”: it uses many different parts simultaneously, harnessing billions of neurons whenever we talk or listen to words. “I’m no cognitive scientist, so this is just speculation,” Ferrucci says, but Watson’s approach — tackling a question in thousands of different ways — may succeed precisely because it mimics the same approach. Watson doesn’t come up with an answer to a question so much as make an educated guess, based on similarities to things it has been exposed to. “I have young children, you can see them guessing at the meaning of words, you can see them guessing at grammatical structure,” he notes.

This is why Watson often seemed most human not when it was performing flawlessly but when it wasn’t. Many of the human opponents found the computer most endearing when it was clearly misfiring — misinterpreting the clue, making weird mistakes, rather as we do when we’re put on the spot.

During one game, the category was, coincidentally, “I.B.M.” The questions seemed like no-brainers for the computer (for example, “Though it’s gone beyond the corporate world, I.B.M. stands for this” — “International Business Machines”). But for some reason, Watson performed poorly. It came up with answers that were wrong or in which it had little confidence. The audience, composed mostly of I.B.M. employees who had come to watch the action, seemed mesmerized by the spectacle.

Then came the final, \$2,000 clue in the category: “It’s the last name of father and son Thomas Sr. and Jr., who led I.B.M. for more than 50 years.” This time the computer

pounced. “Who is Watson?” it declared in its synthesized voice, and the crowd erupted in cheers. At least it knew its own name.

Clive Thompson, a contributing writer for the magazine, writes frequently about technology and science.

Computers Solve Checkers—It's a Draw

King me! Top computer scientist proves perfect play leads to draw, recounts battle for world championship, gets kinged

By [JR Minkel](#)



Image 1



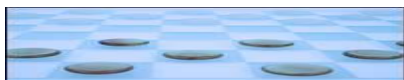
Image 2



Image 3



Image 4



CHECKERS IS SOLVED: Eighteen years of number crunching concludes that perfect checkers ends in a draw. Image: © ISTOCKPHOTO/CHRISTOPHER O DRISCOLL

[Pin it](#)



•

[The Best Science Writing Online 2012](#)

Showcasing more than fifty of the most provocative, original, and significant online essays from 2011, The Best Science Writing Online 2012 will change the way...

[Read More »](#)

Jonathan Schaeffer's quest for the perfect game of checkers has ended. The 50-year-old computer scientist from the University of Alberta in Edmonton left human players in the dust more than a decade ago after a trial by fire against the greatest checkers champion in history.

And now, after putting dozens of computers to work night and day for 18 years—*jump, jump, jump*—he says he has solved the game—*king me!*. "The starting position, assuming no side makes a mistake, is a draw," he says.

Schaeffer's proof, described today in *Science* (and freely [available here for others to verify](#)), would make checkers the most complex game yet solved by machines, beating out the checker-stacking game Connect Four in difficulty by a factor of a million.

"It's a milestone," says Murray Campbell, a computer scientist at IBM's T. J. Watson Research Center in Hawthorne, N.Y., and co-inventor of the chess program Deep Blue. "He's stretched the state of the art."

Although technological limits prohibit analyzing each of the 500 *billion billion* possible arrangements that may appear on an eight-by-eight checkerboard, Schaeffer and his team identified moves that guaranteed the game would end in a draw no matter how tough the competition.

Like any complicated mathematical proof, the result will have to withstand scrutiny. But "it's close to 100 percent," says computer scientist Jaap van den Herik of Maastricht University in the Netherlands, who has seen the details. "He has never published anything that was not completely true."

Opening Play: Walking a Precipice

Schaeffer's odyssey began in the late 1980s. He had written a top chess program but IBM was on the verge of pouring its far vaster resources into Deep Blue. "I like to be competitive," he says, so he turned his attention elsewhere. "I naively thought I could solve the game of checkers," he recalls. "You can teach somebody the rules in a minute."

Setting out in 1989 with 16 megabytes of computer memory, he quickly found that checkers, like chess, was too rich with possible positions to dash off a solution. So he switched gears, vowing to topple legendary checkers champion Marion Tinsley, who had lost only three games in tournament play since 1950.

In 1992 Schaeffer's program Chinook [took on Tinsley](#), who had resigned as world champion when the American Checker Federation and English Draughts Association temporarily refused to sanction the man-computer matchup.

Tinsley was so good that his opponents played dull games in the hope of securing at least a draw, according to Schaeffer; Chinook apparently put the magic back in the game for the champ. "It played brash, aggressive moves—it walked on the edge of a precipice," Schaeffer anthropomorphizes. "It would do things people looked at and said, 'Man, is that program crazy?'"

The program actually beat Tinsley twice, but computer glitches led to a forfeit that gave the human a 3–2 lead with two games left in a best-of-40 match. Schaeffer set Chinook on an aggressive course to try to recoup, resulting in another loss for the computer that cost it and its creator the match, Schaeffer recounted in his book *One Jump Ahead*.

In a rematch two years later, Tinsley withdrew suddenly after six drawn games, citing health problems, making Chinook the winner of the Man–Machine World Championship by default. The checkers champ was diagnosed with pancreatic [cancer](#), which killed him less than a year later.

Chinook soundly beat the top human player in 1996, but Schaeffer's memories of those years are bittersweet. "With hindsight, maybe winning was more important to me at that time than other things. I was a bit obsessed," he says. "My wife would say more than a bit obsessed."

Endgame: King Me!

Early this decade he decided that improved computer speed and memory had brought his goal of solving the game within reach. The first of the solution's two pieces builds on work begun during the Tinsley days. The Alberta researchers exhaustively checked the final stages of play for any arrangement of 10 pieces or less, identifying which of the 3.9×10^{13} positions won, lost or drew.

Next they identified 19 representative opening sequences and searched through subsequent moves for the easiest-to-find connections to final positions—win, lose or draw. To save time, they ignored pointless back-and-forth moves or those that did not turn a draw into a win.

If the sequence of possible moves branches like a tree, the group found that main branch after main branch led to a draw. On April 27, Schaeffer says, the program traced the final branches, completing his 18-year task. He had received his king.

The result itself was not surprising. "We expected that it should be a draw, because at the highest levels there are many games that end in a draw," Herik says.

What the proof does, Watson's Campbell says, is highlight the power of methodically checking outcomes one by one. The most robust chess programs such as Deep Blue and Deep Fritz have beaten world champions by combining swift number crunching with rules of thumb derived from years of human play.

The checkers solver is comparatively dumb, Campbell notes. "All it knows is this move wins or this move doesn't," he says. "It can't 'explain' what it knows."

For that reason, Herik says the results won't be much help to players until researchers can figure out how to translate them into rules or procedures the human mind can follow—a major challenge in itself.

Next Move?

So what game will fall to computers next? Schaeffer and colleagues speculate it will be Othello, an eight-by-eight disk-flipping game. Chess presents a far mightier challenge, but researchers are "in the realm of thinking about" solving it, Herik says, which he calls "a tremendous achievement."

The great white whale of games remains the Asian pastime Go. Although programs have [recently become competitive with grand masters](#) on nine-by-nine boards, they remain toothless on the full 19-by-19 game.

Schaeffer, a true game lover—his two dogs are named Scrabble and Rummicub—intends to pit [his poker program](#) against two top players next week in a 500-hand match designed to minimize the role of chance.

He has come a long way since being fixated on the world championship. Whatever games lie ahead, he can take satisfaction in his current victory. "Eighteen years later," he says, "I'm finally done."



Chunka Mui, Contributor

I help design and stress-test innovation strategies

LEADERSHIP | 1/22/2013 @ 10:12AM | 321,493 views

Fasten Your Seatbelts: Google's Driverless Car Is Worth Trillions (Part 1)

Part One of a Seven-Part Series

Much of the reporting about Google's driverless car has mistakenly focused on its science-fiction feel. While the car is certainly cool—just watch the video



below about a 95%-blind man running errands—the gee-whiz focus suggests that it is just a high-tech dalliance by a couple of brash young multibillionaires, Google founders Larry Page and Sergey Brin.

In fact, the driverless car has broad implications for society, for the economy and for individual businesses. Just in the U.S., the car puts up for grab some **\$2 trillion a year in revenue and even more market cap**. It creates business opportunities that dwarf Google's current search-based business and unleashes existential challenges to market leaders across numerous industries, including car makers, auto insurers, energy companies and others that share in car-related revenue.

Because people consistently underestimate the implications of a change in technology—are you listening, Kodak, Blockbuster, Borders, Sears, etc.?—and because many industries face the kind of disruption that may beset the auto industry, I'm going to do a series of blogs on the ripple effects that the driverless car may create. I'm hoping both to dramatize the effects of a disruptive technology and to illustrate how to think about the dangers and the opportunities that one creates.

In this installment, I'll start the series with a broad-brush look at the far-reaching changes that could occur from the driver's standpoint. In the next installment, I'll show just how far the ripples will reach for companies—not just car makers, but insurers, hospitals, parking lot operators and even governments and utilities. (Fines drop when every car obeys the law, and roads don't need to be lit if cars can see in the dark).

After that, I'll explore how real the prospects are for driverless cars. (Hint: The issue is when, not if—and when is sooner than you think.) In the last three installments, I'll go into the strategic implications for Google, for car makers and, finally, for every company thinking about innovation in these fast-moving times.

To begin:

Driverless car technology has the very real potential to save millions from death and injury and eliminate hundreds of billions of dollars of costs. Google's claims for the car, [as described by Sebastian Thrun](#), its lead developer, are:

1. We can reduce traffic accidents by 90%.
2. We can reduce wasted commute time and energy by 90%.
3. We can reduce the number of cars by 90%.

To put those claims in context:

About [5.5 million motor vehicle accidents](#) occurred in 2009 in the U.S., involving 9.5 million vehicles. These accidents killed 33,808 people and injured more than 2.2 million others, 240,000 of whom had to be hospitalized.

Adding up all costs related to accidents—including medical costs, property damage, loss of productivity, legal costs, travel delays and pain and lost quality of life—the American Automobile Association studied crash data in the 99 largest U.S. urban areas and [estimated the total costs to be \\$299.5 billion](#). Adjusting those numbers to cover the entire country suggests annual costs of about \$450 billion.

Now take 90% off these numbers. Google is claiming its car could save almost 30,000 lives each year on U.S. highways and prevent nearly 2 million additional injuries. Google claims it can reduce accident-related expenses by at least \$400 billion a year in the U.S. Even if Google is way off—and I don't believe it is—the improvement in safety will be startling.

In addition, the driverless car would reduce wasted commute

Great Mileage

Some Benefits of the Driverless Car

time and energy by relieving congestion and allowing cars to go faster, operate closer together and choose more effective routes. [One study](#) estimated that traffic congestion wasted 4.8 billion hours and 1.9 billion gallons of fuel a year for urban Americans. That translates to \$101 billion in lost productivity and added fuel costs.

The driverless car could reduce the need for cars by enabling efficient sharing of vehicles. A driverless vehicle could theoretically be shared by multiple people, delivering itself when and where it is needed, parking itself in some remote place whenever it's not in use.

A car is often a person's second largest capital expenditure, after a home, yet a car sits unused some 95% of the time. With the Google car, people could avoid the outlay of many thousands of dollars, or tens of thousands, on an item that mostly sits and, instead, simply pay by the mile.

A [study](#) led by [Lawrence Burns](#) and William Jordon at Columbia University's Earth Institute [Program on Sustainable Mobility](#) showed the dramatic cost savings potential. Their analysis found that a shared, driverless fleet could provide far better mobility experiences than personally owned vehicles at far radically lower cost. For medium-sized cities like Ann Arbor, MI, the cost per trip-mile could be reduced by 80% when compared to personally own vehicles driven about 10,000 miles per year—without even factoring in parking and the opportunity cost of driving time. Their analysis showed similar cost savings potential for suburban and high-density urban scenarios as well.

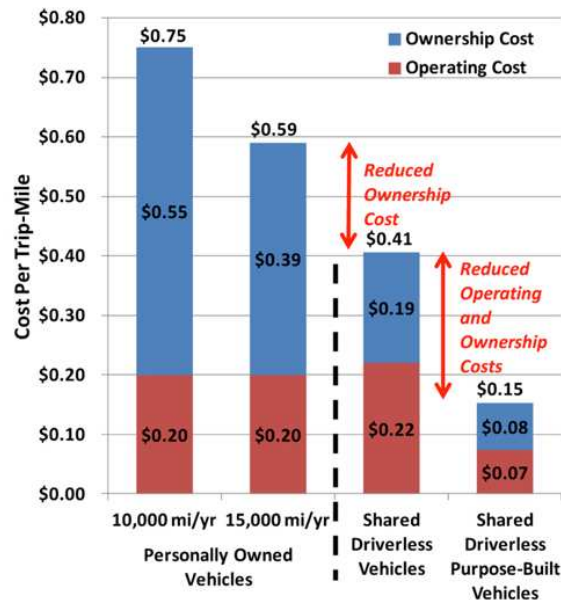
Some Benefits of the Driverless Car

Google's Aspiration	Potential Annual Benefits (US only)
<ul style="list-style-type: none"> 90% reduction in accidents 	<ul style="list-style-type: none"> 4.95 million fewer accidents 30,000 fewer deaths 2 million fewer injuries \$400 billion in accident-related cost savings
<ul style="list-style-type: none"> 90% reduction wasted commuting 	<ul style="list-style-type: none"> 4.8 billion fewer commuting hours 1.9 billion gallons in fuel savings \$101 billion saved in lost productivity and fuel costs
<ul style="list-style-type: none"> 90% reduction in cars 	<ul style="list-style-type: none"> Reduce cost per trip-mile by 80% or more Increase car utilization from 5-10% to 75% or more Better land use

Sources: Google, US NHTSA, AAA, Texas A&M Transportation Institute, Columbia University Earth Institute and Devil's Advocate Group's analysis

Shared Driverless Fleets

Personal travel costs can be dramatically reduced



Source: Program on Sustainable Mobility,
The Earth Institute, Columbia University

Driving could

become Zipcar writ large (except the car comes to you).

Looking worldwide, the statistics are less precise, but the potential benefits are even more startling. The World Health Organization [estimates](#) that more than 1.2 million people are killed on the world's roads each year, and as many as 50 million others are injured. And the WHO predicts that the problems will only get worse. It estimates that road traffic injuries will become the fifth leading cause of worldwide death by 2030, accounting for 3.6% of the total—rising from the ninth leading cause in 2004, when it accounted for 2.2% of the world total.

If Google could give everyone a world-class electronic driver, it would drastically reduce the deaths, injuries and direct costs of accidents. The driverless car might also save developing countries from ever having to replicate the car-centric infrastructure that has emerged in most western countries. This leapfrogging has already happened with telephone systems: Developing countries that lacked land-line telephone and broadband connectivity, [such as India](#), made the leap directly to mobile systems rather than build out their land-line infrastructures.

China alone expects to [invest almost \\$800 billion on road and highway construction](#) between 2011 and 2015. It is doubtful, however, whether even this massive investment can keep up with the rising accidents and traffic congestion that the country endures. And road construction won't deal with the issue of pollution, to which the massive car buildup contributes and which, as the following FT video reports, is becoming an ever more politically sensitive issue.

How might China and other developing economic powers' massive car-related investments be redeployed if fundamental assumptions were viewed through the lens of the driverless car?

In sum, the Google driverless car not only makes for a great demo; it has worldwide social and economic benefits that could amount to trillions of dollars per year.

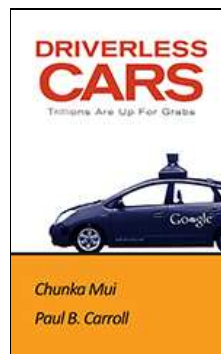
* * *

There is an old saying: One man's savings are another man's lost revenue. In [Part 2 of this series](#), I explore the enormous business threats and opportunities that will stem from Google's driverless car.

— — —

This series draws from research for [The New Killer Apps: This Time, Incumbents Can Beat Startups](#), a forthcoming book coauthored with Paul B. Carroll. To learn more, visit the [book's Facebook page](#).

An ebook, [Driverless Cars: Trillions Are Up For Grabs](#), which integrates the seven columns of this series, incorporates many of the comments and adds our latest research, is available at [Amazon](#).



Forbes Series Outline:

1. [Fasten Your Seatbelts: Google's Driverless Car Is Worth Trillions](#)
2. [The Ripple Effects—As Far As The Eye Can See](#)
3. [Why Change Will Come Sooner Than You Think](#)
4. [How Google Wins](#)
5. [How Automakers Still Win](#)
6. [Will Auto Insurers Survive Their Collision with Driverless Cars?](#)
7. [Driverless Cars Are Just One of Many Looming Disruptions](#)

Follow Chunka Mui here at Forbes (click on the +follow button next to his picture), on Twitter [@chunkamui](#) or at [Google+](#).

This article is available online at:

<http://www.forbes.com/sites/chunkamui/2013/01/22/fasten-your-seatbelts-googles-driverless-car-is-worth-trillions/>

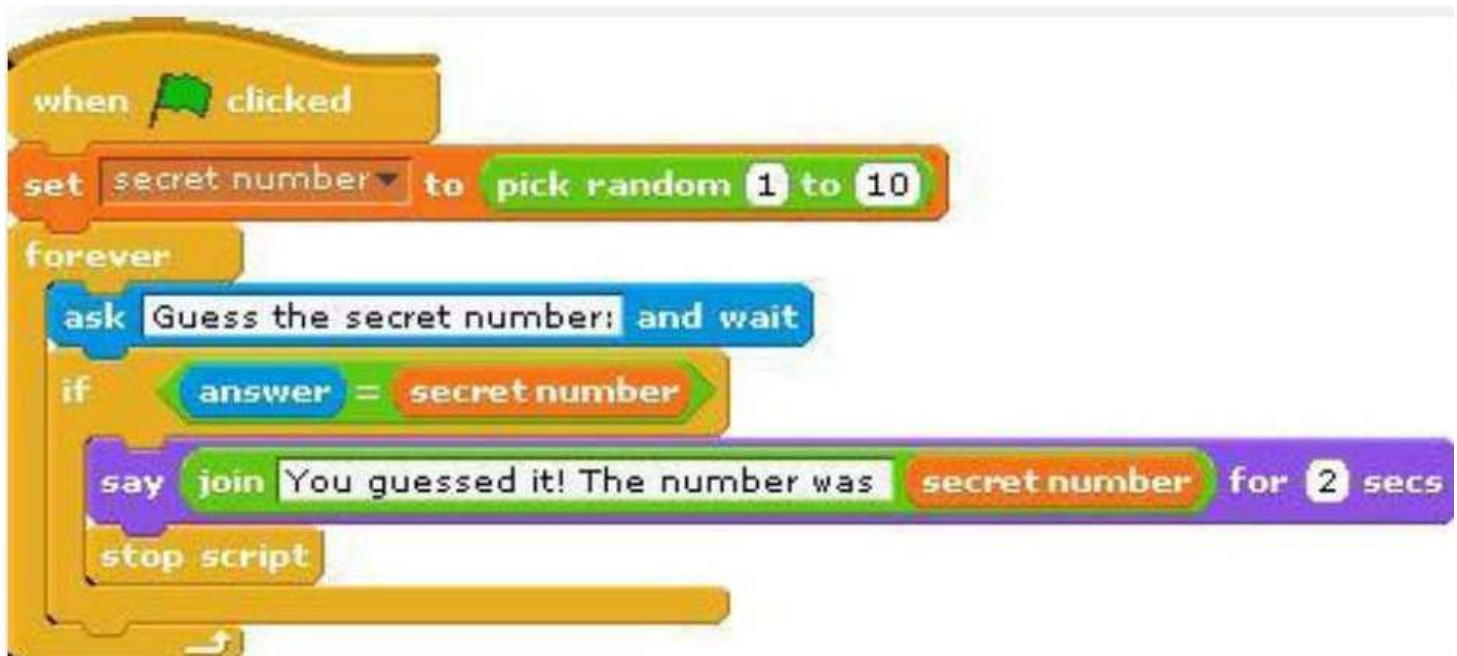
Lab: Random Numbers and Conditionals

In this lab you will create a guessing game. Look at the script below. The computer chooses a random number and then asks the player to guess the random number.

Algorithm

```
When flag clicked
  Set the secret number 1 to 10
  Repeat forever
    Ask for guess and wait
    If the answer = the secret number
      Say Guessed It Message
    Stop the loop
```

Script



Step 0

Before beginning to build your program, let's look at what variables we will need to begin the game.

- The secret number
- The guess from the player

Next, we need to figure out the tasks that the sprite and player will be enacting.

- The Sprite will begin by setting up the game's variables and choosing the secret number.

- The Sprite will ask for a guess from the player.
- The player will input their guess.
- If the player's guess was correct, the sprite will give the player a message and the game will end.

Note that there is a new block above that will return a random number between the two input numbers (inclusive). This block is in the Operator's palette. You can enter a number or use a variable. Because this is a reporter block you can use it inside other blocks as an input.



The game is not that cool yet, though. In the next few steps, we are going to make the game much more like playing the number-guessing game with a person! Here are tasks to try to make the code act more like a human! If you think of anything else that would make it cooler, feel free to add it in. Try each of these in order. When you are done, your finished code should be able to deal with all of the tasks.

Step 1

Add to your beginning script the blocks so that the sprite welcomes the player and asks for their name before beginning.



Step 2

We want to give the player more information if they don't guess correctly. Have the sprite tell the player if the secret number is bigger or smaller than the number that they guessed.

Step 3

Right now, the sprite always picks a number between 1 and 10. Change this so that the sprite always picks a number between 1 and a variable named max. (Don't forget to add the variable.) Add to your script so that it asks the player what they would like the maximum number to be, before choosing a random number. Use this maximum number as the highest number that the sprite will choose.

Step 4

Now let's keep track of how many guesses it takes before the player guesses the right number. You will need a new variable for this task and will need to add to it every time a guess occurs.

Step 5

When the player guesses the secret number, tell them how many guesses it took, and congratulate them using their name.