*Published at:* https://www.format.com/magazine/features/design/you-have-died-of-dysentery-oregon-trail-design

# I Designed *The Oregon Trail*, You Have Died of Dysentery

The lead designer of The Oregon Trail explains his team's process for creating the most successful educational computer game of all time.

R. Philip Bouchard

24 July 2017

*The Oregon Trail* is, by most measures, the most successful education computer game of all time. Many versions of the game were created over the years, dating all the way back to 1971. However, the version that many people assume is the “original”—the version that constantly appears in memes and magazine articles is the 1985 Apple II version. (The 1985 design was also implemented on the IBM PC, Macintosh, and other platforms, but it first appeared on the Apple II.) I was the lead designer and team leader of this most famous version of this most famous educational game, and therefore I am an excellent source for revealing exactly how this design came to be.

Because this is a success story—and “success has many fathers”—the names of several other people will appear in this account (not just my own name, despite the braggadocio in the title of this piece). All of these people played key roles in the success of *The Oregon Trail*.

There are several reasons behind the popular impression that the 1985 version was the “original”. One obvious reason is that the 1971 version was text-only, and relatively few people have seen or played the text-only version. A second big reason is that most of the features that people fondly associate with the game were first introduced in 1985. The 1985 product was far more than a new version—it was a complete reimagining of how to create a game about the Oregon Trail. I am quite certain that the 1985 design was the most original and creative version in the entire history of the product, excluding (of course) the completely original 1971 version. In other words, the game that ultimately became so successful was invented in two equally important stages—one in 1971, and the second in 1984–85.

## The True Original Version

The very first version of *The Oregon Trail* began as a concept in the head of Don Rawitsch, who in 1971 was a student teacher in Minnesota. He imagined a board game that could be played by students in the class that he was scheduled to teach. Soon after Don began work on this concept, he shared his idea with his roommates—Bill Heinemann and Paul Dillenberger—who were also student teachers. They were enthusiastic about Don’s idea, but they suggested that it should be a computer game instead of a board game. The Minneapolis school district, where all three taught, had recently purchased a mainframe computer, and each school in the district had a teletype terminal for remote access to the computer. Furthermore, both Bill and Paul had learned BASIC, a simple programming language that was ideal for creating a small computer program such as the one they envisioned.

Don quickly agreed to the change in plans, and the three began to design and program the concept. Two weeks later, just in time for the class that Don was scheduled to teach, a first rough version of the game was ready to show the kids. Despite the awkwardness of the teletype machine, the kids clearly enjoyed playing the game. Bill and Paul observed similar results at their school. The game remained available, accessible to students in their spare time, for the rest of the semester. But at the end of the term, when his period as a student teacher came to a close, Don deleted the program from the computer system.

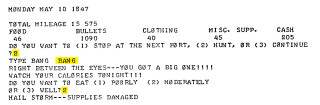
A few years later Don began work at MECC (the Minnesota Educational Computing Consortium), a new Minnesota state agency tasked with providing computing access to all of the schools in the state. MECC’s computing model was similar to that of the Minneapolis school district—a centrally located mainframe computer that could be accessed via a teletype terminal in each school. But now, instead of serving a single school district, this new organization served the entire state. Don had saved a printout of the BASIC program from the 1971 game, so in 1974 he typed the same program into MECC’s computer system. Then he did some fine-tuning by tweaking the frequency of the various random events in the game. Finally, in 1975, he made the game available to all of MECC’s users. OREGON (as it was called) soon became the most popular educational activity on the system—and it remained so until MECC shut down its mainframe operations in 1983.

This early text-only version of *The Oregon Trail* was exceedingly simple—as were nearly all of the educational activities on the MECC mainframe. And yet seven of the core concepts that appeared in the 1971 game have re-appeared in every version since then.

These core concepts are:

* The player buys supplies before starting the journey to Oregon.
* There are opportunities to hunt for food along the way.
* There are opportunities to make purchases at forts along the route.
* The player must manage the level of supplies to avoid running out.
* The rate of travel depends upon the current conditions.
* Misfortunes frequently occur.
* The game ends when the player reaches Oregon, or when the player dies along the way.

The 1971 game was structured as a repeating cycle. Each cycle represents two weeks of travel on the trail. Below is an example of a typical cycle, from an actual session of the game. Most of the text in this example was typed by the computer, but at times the computer stops to wait for a response from the player. The player’s responses are highlighted in yellow:



As you can see from this example, each cycle begins with a status update. Then the player decides whether to hunt. (Sometimes the option to visit a fort is also available, as shown here.) If the player chooses to hunt, then he shoots the rifle by typing “BANG” when told to do so. If the typing was fast and accurate, then the hunting is successful. The player now decides how much to eat. Finally, one or more “misfortunes” occur, which can reduce the supplies. Then the cycle repeats two weeks later, taking into account supplies used in the interim. The cycling continues until the player either reaches Oregon or dies trying. A successful journey requires approximately 12 cycles—depending upon various circumstances that affect the rate of travel.

For many players, the principal appeal of this game was the challenge of hunting—typing the word “BANG” fast enough to be successful, without making a typing mistake. Therefore the typical way to play was to give an identical response in every cycle—first type “2” to hunt, then type “BANG,” then type “2” to eat moderately. The only necessary deviation is when one of the supply items gets dangerously low, forcing the player to go shopping at a fort instead of hunting. The second appealing aspect of this game was the challenge of managing the resources—keeping an eye on the supplies to ensure that none of them run out, which would be a fatal mistake. The third appealing aspect is that there are so many different misfortunes that might occur. So you never know what will happen next—unless you run out of any supplies, in which case you will probably die quite soon.

The original design also featured random attacks from wild animals, bandits, and “hostile riders,” each of which provided additional opportunities to type “BANG”—and dire consequences for failing to type rapidly and accurately. In the late 1970s a small change was made to the game, so that the player would not know in advance what word he would be expected to type when it came time to shoot. Sometimes it would be “BANG,” but it could just as likely be “POW,” “BLAM,” or “WHAM.”

## The World Changes

In 1978 Don Rawitsch published the BASIC code for OREGON in *Creative Computing* magazine. After that many people tweaked the code to run on various brands of microcomputers. Someone—I don’t know who—put a version of this program code on an Apple II. This version of the game was not a new design—in fact it mostly used the same BASIC code that appeared in the magazine. However, the author did make three important changes to the game, one of which was a reimagining of the hunting activity. Instead of being text-based—like all other parts of the game—this new activity presented a sketch of a deer, which the player could shoot at by pressing any key on the keyboard.

By 1979, MECC was buying Apple II computers in large quantities (at a discounted price), and reselling them at cost to schools all over Minnesota. In 1980 MECC also began to distribute disks containing Apple II software to these same schools. (These disks were provided for free to Minnesota schools, but sold at a significant profit to schools outside Minnesota.) Each disk contained a collection of programs ported over from MECC’s mainframe computer. The OREGON game appeared in two of these collections, one of which was named “Elementary Volume 6[.]”

MECC continued to distribute “Elementary Volume 6” for many years, and the game OREGON remained MECC’s most famous product, even after MECC shut down its mainframe operations to focus entirely on microcomputers (Apple II, Atari, IBM, etc.). However, the competitive landscape completely changed between 1980 and 1984. In 1980, all educational computer software was still being designed and programmed by amateurs—hobbyists and teachers who created these small, simple programs in their spare time. Furthermore, no large private companies had yet entered the educational software market—neither the school market nor the home market.

By 1984 the world of educational software was completely different. A great number of companies—both large and small—had begun to sell educational computer software. Some of these companies specialized in the school market, while others specialized in the home market. By 1984, all of the best educational software was designed and programmed by professionals—people whose full-time job was to design or program such software. In order to remain competitive, the same transition occurred within MECC. By 1983 MECC had acquired a staff of professional designers and programmers tasked with creating top-quality original software for the Apple II. By 1984 the improvement in MECC’s products was quite obvious to everyone. This was the beginning of MECC’s “golden age”—a decade-long period in which MECC turned out exceptional new software year after year—mostly targeted to the school market, but also including a few titles for the home market.

Although MECC was now focused almost entirely on the creation of new, original titles, it also had three famous simulations from the 1970s—*Oregon Trail, Lemonade Stand,* and *Odell Lake*—that it did not want to abandon. All three activities had been ported to the Apple II by 1980, but by 1984 they were embarrassingly outdated—especially in terms of their appearance. MECC decided that it would create new versions of all three activities.

## The Decision to Break with the Past

It was in this context that I was chosen by MECC in October 1984 to lead the creation of a new version of *The Oregon Trail*. I had joined MECC’s staff in 1981 with a substantial background in the design and programming of computer-based simulations (although prior to 1981 all of my simulations had been targeted to college-level students). I was the only person on staff with this kind of experience, and therefore I was the logical choice for the role of lead designer and team leader. Still, I was surprised by the details of the mandate that I was given.

First of all, I was told to design a product for the home market, not the school market. Previously, MECC had always specialized in the school market. This would be MECC’s very first attempt to create a product specifically designed for the home market.

Second, my instructions were to do far more than just create a better-looking version of the original game. I was told to take the original concept and run with it. I was to expand upon the original concept, building a much more elaborate and robust game. I could do so in any way that I saw fit, provided that I preserved whatever magic had made the original so popular. However, I had to design the product specifically for the Apple II—which resulted in some serious restrictions, such as having to work with a palette of just six colors. (I would have preferred to use a more recent machine, such as the Commodore 64.)

My mandate to re-envision and redesign *The Oregon Trail* was almost overwhelming at first—the possibilities were endless, yet I had to get it absolutely right on the first release. For 13 years, from 1971 to 1984, the OREGON game had remained essentially unchanged. A few small details had been tweaked along the way, but never had the product been completely re-imagined and redesigned. Never had the underlying models been changed—the structures, algorithms, and assumptions upon which the game is based. For the very first time, we were going to throw out everything—including all of the existing software programming, which dated back to 1971—and start completely from scratch. Every detail was up for reconsideration. Furthermore, I needed to create a much richer and more elaborate experience than the original OREGON—and this would require a great deal of new, original thinking.

To kick off the project, I defined a key metric to serve as the cornerstone of our approach: “Do the kids that love the old version like the new version even more?” The idea was to perform this test periodically throughout the project, to verify that we were still on track. Whenever we found ourselves deep in the weeds, this yardstick would bring us back to reality, allowing us to see the big picture again.

Because I was designing a home market product, I knew that I had to create a game that was highly entertaining, in addition to being clearly educational. I felt that it was possible to do both, without seriously compromising either—but I had to strike a careful balance in the details. In particular, I felt that both the educational value and the entertainment value should arise from immersing the player in a historically accurate experience.

From the very beginning, I saw that this project presented many challenges, but the issue of space constraints seemed especially challenging. The new game needed to have a rich set of color graphics, but the graphics would require a lot of space—both on the floppy disk and in the Apple II RAM (memory). I also hoped to add many new details and gameplay aspects, all of which would require space. Unfortunately, I was designing for a computer that had only 64K of RAM. Furthermore, the product would be distributed and played on a double-sided 5.25” floppy disk—providing a total of just 280K of storage space. (Note that the Apple II did not have a hard drive, only a floppy disk drive.)

As the project began, our core team consisted of five people. In addition to me (Philip Bouchard), there was also John Krenz (lead programmer), Charolyn Kapplinger (lead artist), Shirley Keran (research), and Bob Granvin (additional programming). All five of us played active roles in the early brainstorming and planning for the product, although MECC eventually moved Bob to a different project that needed his attention. And so the five of us began a project that became MECC’s biggest effort to date, lasting a total of 10 months, from October 1984 to the end of July 1985.

## Developing the Key Concepts

In MECC’s traditional design approach (as of 1984), the instructional designer first imagines a computer-based student activity, and then creates a series of sketches that illustrate what should appear on the screen in each of the key steps of the activity. This collection of sketches is then packaged with written notes that provide additional details. The sketches and notes together constitute a “design document.” The designer then reviews the document with a small team of up to three other people—the lead programmer, the lead graphic designer, and an experienced programmer/analyst. Together they identify missing details in the design, and they help to resolve any other issues that the design presents. The designer then makes any additions or changes that are needed to complete the design, and the team gets to work creating the graphics and generating the program code.

This model worked well enough for the relatively simple products that MECC had been creating up to that time, but I could see that it would not work for the product that I intended to design. First of all, I envisioned the game as being based on a complex interlocking set of mathematical simulation models—a weather model, a health model, a travel model, and so on. Furthermore, each of these models would be based on a complex interlocking set of mathematical formulas. These models would require incremental design and tuning—which meant not only that I needed to crank out a great number of formulas, it also meant that the design needed to be specified and implemented in iterative stages, instead of all at once.

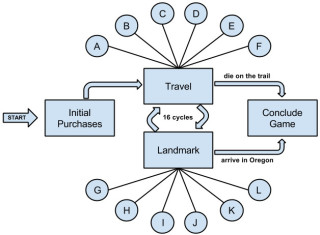
Second, I felt that the best way to build an innovative but successful design was to start with a an ambitious set of new ideas, and then to gradually build and test these ideas through successive refinement—first by creating a simple prototype, and then by building up the real product one aspect at a time. At each step of the process we should evaluate where we are, compared to where we want to be, and then prioritize what to do next. Some of our cherished ideas would have to be shed along the way, but by constantly prioritizing and improving, we would achieve a successful result in the end. (I first learned and applied the concepts of successive refinement when I studied computer science in graduate school in the 1970s. However, decades later, I saw a lot of overlap between these ideas and the concept of “agile programming.”)

But first, I need to design the underlying framework for the game, and for a couple of weeks this had me stymied. The original OREGON was based on the concept of “turns,” each representing exactly two weeks of real time. This simple framework was well-adapted to its original purpose—a text-based game played on a teletype—but it was poorly adapted to the needs of the new game. I wanted to create a framework that was intimately tied to the actual geography of the Oregon Trail, and I wanted to give the player a much greater set of opportunities to make decisions—not just what to buy and how much to eat. It soon became clear that I would have to invent a completely different framework for the new game. But as I considered one idea after another, none of these new ideas seemed to work.

So I broke the problem into pieces, and I began to tackle each piece individually. Eventually, over a period of several weeks, I was finally able to generate a solution to every problem. The resulting new framework for *The Oregon Trail* was based on seven key concepts, none of which were present in the 1971 design:

1. To tie the new design to the actual geography of the Oregon Trail, the journey will consist of distinct segments of varying length, each ending at a unique major landmark along the trail. (Of all the many differences between the 1971 product and the 1985 design, this might be the most important — in part because it enabled so many other crucial changes.)
2. A set of activity modules is available at each landmark. In many cases there is a key activity tied to the landmark, such as crossing a river or purchasing goods at a fort.
3. The landmarks are grouped into distinct geographic zones, each with its own data about climate, terrain, and wild animals.
4. Most of the activity modules are data-driven and contain randomization, providing a different experience each time that the module is used. For example, each time that the hunting activity is launched, it uses the zone data in order to present animals and terrain that are appropriate to the current location on the trail. But even if the player hunts in the same zone over and over, the arrangement of the terrain will always be unique, and the animals will appear at unpredictable times and locations.
5. In addition to the landmark-to-landmark cycle, there is an independent daily cycle, which includes all of the resource management computations (supplies, health, etc.) along with the random events. Therefore the core computational cycle in the 1985 design is one day, instead of the two-week cycle used in the original design. This has huge ramifications for all the computations and all the interactions.
6. Between landmarks, the journey proceeds automatically, but the user can pause at any time. (This was an especially crucial concept, because it provides the user with access to many activities and events, without constantly interrupting the game.) However, the journey pauses automatically if a major event occurs.
7. Pausing between landmarks provides access to another full set of activity modules, some of which are distinct from the activities available at landmarks. (For example, you can talk to people at landmarks, and you can go hunting between landmarks.)

This new framework was dramatically different from the 1971 design, allowing me to create a much richer experience for the player. In the new design, the relationship between the activity modules looked like this:



With this highly flexible structure serving as backbone of the new product, we could now plan what activities would be accessible while traveling between landmarks (A-F in the diagram) and what activities would be accessible at each of the landmarks (G-L in the diagram). Each of these distinct modules (represented by the four rectangles and the 12 circles) could be designed and programmed separately. Furthermore, each of the half dozen mathematical simulation models (not shown in this diagram, but operating underneath it all) could also be designed and programmed separately.

My teammates were enthusiastic about this new framework, so I began the long process of fleshing out the details. First of all, based on my research and that of my colleague Shirley, I prepared a list of candidate landmarks. Then, by applying several selection criteria, I trimmed the set down so that the journey would consist of approximately 16 legs. I also introduced the concept of “cutoffs,” meaning that in addition to the principal route, the player would sometimes have the option to take an alternate route. Then I began work on creating a list of possible activity modules, systematically turning out “concept documents” for all of the most promising ideas.

Throughout this concept development stage, I conferred heavily with my teammates. Before writing up each concept document, I would usually discuss the ideas I had with my teammates. And then after I wrote each concept document, I would review the document with my teammates to get their additional feedback. The upshot is that even though I was the author of all but one of the concept documents, the ideas contributed by my teammates (Cheryl, John, Shirley, and Bob) were often reflected in my documents. This was especially true of the initial concepts for the hunting activity, which Bob, John, and I hashed out together.

## Incorporating Humans into the Design

Early in the project, my second highest priority—second only to incorporating real geography in the product—was to include human characters in the design. The 1971 design omitted the concept of people, and therefore you travel the entire 2000-mile route without ever meeting or interacting with another person. I wanted to change that. I brainstormed many distinct concepts for how the player might interact with human characters along the way—or at least be aware of other humans—and I sincerely hoped to include nearly all of these ideas in the finished product. But as the project progressed, I had to cut some of my cherished concepts, because of limited space on the floppy disk and limited budget to design and build the product. However, I was able to incorporate four of the concepts into the design, each of which made a huge impact on the “human feel” of 1985 product:

* Before beginning the journey, you name four other people who will travel with you—typically family members or friends. Each member of your 5-person party is subject to illness and accidents. As you travel the trail, a principal goal is to keep all of these people alive and healthy. If any of them gets sick or dies, then they are mentioned by name. This collection of concepts was a crucial addition to the new design.
* As you purchase your goods before starting the journey, you enter a store (“Matt’s General Store”) and interact with the proprietor of the store. Matt provides helpful advice as you make your purchases.
* On the real Oregon Trail, people tended to congregate at each of the landmarks. These people included not only other travelers, but also Native Americans, local traders, and soldiers. Therefore, at each landmark in the game, the player can meet and talk to three different people—each of whom offers an interesting perspective on his or her experiences. Furthermore, many of these conversations provide helpful hints about how to survive the journey. (Unfortunately, due to limited disk space, we could not include pictures of these characters.)
* If you make it all the way to Oregon, then the high-score list is pre-populated with the names of actual people who made the journey to Oregon or who served as early explorers in the region.

Thanks to the lead graphic designer, Charolyn Kapplinger, people also appear in many of the large landmark graphics. This also adds significantly to the human side of the design.

My single biggest regret regarding *The Oregon Trail* is that I had to abandon most of my ideas for complex, nuanced interactions with Native Americans. Several such modules were included in the design concepts that I wrote up. However, a few of the simpler ideas did make it into the finished product. For example, you are much more likely to have a successful crossing of the Snake River if you hire a local Indian to guide you—which was also the case on the real Oregon Trail.

## Other Top Priorities

Another top priority for me was to include river crossings—a key aspect of the Oregon Trail experience that was missing from the original design. I chose a set of representative crossings—a small subset of the actual number of crossings made by the travelers on the trail. At each river the player must choose between several methods of crossing, taking into account the current conditions at the crossing. A highly detailed simulation model underlies this module, controlling the current conditions of the river (based on the recent weather, among other factors) and also controlling the probabilities of various outcomes, based on conditions and the player’s choice of how to cross. The results are communicated in a nail-biting animation that puts many players on the edge of their seat.

Still another of my priorities was to include two more “action” activities, in addition to hunting. I soon modified this plan so that one of these activities would operate as an extended logic puzzle, rather than an action game. These two activities would be associated with the final leg of the journey to Oregon, the last 100 miles. The player could choose to raft down the Columbia River (the action activity), or to take the Barlow Toll Road, in which case the activity is the challenge of getting the wagon up and down through the difficult mountainous terrain along the route. Again, disk space and budget forced us to make cuts—but a stripped-down version of the rafting game did eventually make it into the product.

And finally, one other priority was to find as many ways as possible to build “replayability” into the product. This had seldom been a concern for MECC on any previous product, because all of these products had been targeted to the school market. For schools, where children far outnumbered computers and access time was highly limited, most of MECC’s products were designed so that the full value of the product could be experienced in a brief amount of time. But for the home market, I needed to design a product that a child could visit over and over—and it would still be interesting after 20 or 30 or 40 hours of play. To encourage replay, I incorporated a long list of features and details into the new design:

* The initial motivating factor—identical to the motivator in the original product—is to survive the entire journey to Oregon. As in the original, most players fail on the first attempt—but they perceive that it is possible to do better. And so they keep trying, doing better and better on most tries, until they finally succeed in reaching Oregon.
* Just as the player reaches Oregon for the first time, a surprise is unveiled—by surviving all the way to Oregon, the player has been awarded a score. This score is based on several distinct factors, which are clearly itemized. So at exactly the moment when the player achieves the initial goal—to reach Oregon—a new, highly motivating goal is revealed — to get a much better score.
* The player also sees that the high-score list is pre-populated with names—and most of these people have higher scores than what the player has achieved. This initial list creates a clear benchmark to strive for—to climb up and up the list until finally exceeding the top score on the list (that of the explorer Stephen Meek).
* As the player seeks to get higher scores, it becomes apparent that it is necessary to attempt the more difficult starting conditions—traveling as a carpenter (with less money) or a farmer (with even less money), rather than traveling as a banker. Although it is more difficult to reach Oregon when you have less money, you get more points for your efforts.
* The new game has far more options and alternative approaches to explore than the original game. Over time, many players become interested in exploring these alternatives. What if I start in a different month? What if I change the pace or the rations? What if I stop for rests every now and then? What if I talk to a lot more people to get advice? What if I float across each river instead of trying to ford any of them? What if I hunt less (which uses up a day of time in the new design) and shop more—or vice versa? What if carry a greater number of spare wagon parts? What if I take the Barlow Toll Road instead of rafting the Columbia River—or vice versa?
* As in the original game, some players will want to play over and over just to go hunting. And to that end, I wanted the new hunting game to be both challenging and compelling. All of these factors—and several others that I included in the design—resulted in a game with a very high degree of replayability. Without this characteristic, the game would not have been very successful in the home market.

## The Mathematical Models

In both the original 1971 game and the new 1985 design, the simulation is driven by a set of mathematical formulas tied to state variables. Each state variable tracks the current value of some important quantity. In the 1971 design there were exactly eight state variables:

* Cash (in dollars)
* Food (as a dollar value)
* Ammunition (the number of bullets)
* Clothing (as a dollar value)
* Miscellaneous supplies (as a dollar value)
* Oxen (as a dollar value)
* Distance traveled so far (in miles)
* Current date (incremented by two-week jumps)

Note that there is no state variable for health in the 1971 product. Therefore your health in the current turn does not depend in any way upon your health in the previous turn. The game ends if you die, and you can die from any of the following states:

* You run out of food and starve.
* You run out of “miscellaneous supplies” and you happen to get sick. (Cold weather and insufficient clothing increase the probability of becoming sick.) The rationale was that every time you get seriously sick, you must have medicine or else you will die.
* You run out of money and you get sick. The rationale was that every time you get seriously sick, you must pay a doctor or else you will die.
* You run out of bullets and are attacked by “hostile riders” who massacre you.

I strongly disagreed with the idea behind three of these four end-game conditions—only starving made sense to me. The year in which I set the game (1848) was long before the invention of modern antibiotics. The medicines available at that time had little effect on the diseases commonly contracted along the Oregon Trail. Therefore it made no sense that all diseases should be 100% curable with medicine, and 100% fatal without medicine.

Likewise, it made no sense that you could always find an available doctor anywhere along the trail, and that you would always die without a doctor, but always live if a doctor was present—and that a doctor would never, ever treat you for free. I eventually eliminated medicine and doctors from the design entirely.

I also disagreed with the frequent attacks by bandits, wild animals, and “hostile riders” in the original game—and the need to drive all of them off by shooting at them. None of this was consistent with my historical research regarding the Oregon Trail. So I removed all of these interactions from the design.

For the new design, I completely reimagined how the underlying simulation model should work. Part of this change was necessary because of my switch from two-week time increments to one-day time increments. But most of the change was driven by a desire to provide a much richer model, and also by a completely new vision of what should trigger the deaths of the five members of the party. The new game would include a complex health model that employed several different state variables, allowing the game to track the general health of the party on a daily basis, as well as the state of recovery of each party member from any accidents or diseases.

Another important change was the inclusion of a complex climate and weather model, based on the actual month-by-month values for average temperatures and rainfall at various points along the trail. (In the original game, the month of the year has no effect upon the weather.) As the player travels along the trail, each day’s weather is based on the current month and the player’s current location along the trail. The simulation retrieves the corresponding average temperature and adds or subtracts a random deviation. The simulation also retrieves the odds of rainfall and then generates conditions that may be dry, rainy, or very rainy. (If the weather is very cold, then snow replaces rain.)

Instead of a single simulation model, I designed the following interlocking models, each consisting of several to many interlocking mathematical formulas:

* Climate and weather
* Health
* Progress on the trail
* Supplies (cash, food, oxen, bullets, clothing, and spare parts)
* River crossings
* Scoring system

I ended up with a huge set of state variables, and a huge set of mathematical formulas. With so many details affecting one another, I knew that I would have to do a lot of fine tuning of the gameplay as we tested the game. Especially in the final three months of the product development, I tweaked all of the formulas and starting values in order to produce a better and better balance in the gameplay.

## The New Set of Core Concepts

By the end of July 1985, *The Oregon Trail* was complete and ready for release. To our delight, the product was an instant hit. In addition to becoming quite popular in the home market, it soon became the most ubiquitous educational software product in North American schools. Our new game, along with its successor versions, made millions of dollars for MECC’s owners over the following years. (MECC had begun as a state agency, but it was later sold to private investors.) Although these riches did not trickle down to the people who had designed and built the game, we were quite gratified to see what we had accomplished—the creation of a highly successful product that ultimately became a classic game and a cultural icon.

Designing and building this completely new version of *The Oregon Trail* had not been a straightforward, linear process. In my early design documents, I had proposed and considered all sorts of ideas, only some of which made it into the finished product. By the time we released the game, the following innovative new features had been incorporated:

1. **Real geographic details.** The original OREGON game included almost no geographic details at all. No rivers, mountains, forts, or other landmarks were ever mentioned by name. A key goal of my new design was to incorporate lots of real geography, woven throughout the game, illustrated with attractive color graphics.
2. **Landmark-based travel cycle.** The original OREGON employed a very simple two-week game cycle, without any landmarks. I divided the primary route into 16 segments, each beginning and ending with a famous landmark. At each landmark, and also between landmarks, the player makes many decisions that were not available in the original game.
3. **Continuous daily cycles.** The journey to Oregon typically takes about 150 days. I needed to incorporate a daily cycle into the design, so that crucial events could occur on any day of the journey, and to allow the player to track the supplies on a daily basis. To avoid halting the game over and over, the game continues on “auto pilot” between landmarks, until the player interrupts or a major event occurs.
4. **Choices as to the route.** The original OREGON presented the Oregon Trail as a single continuous path, without any branches or choices of route. I introduced a more realistic situation where the player must occasionally make a decision as to which way to go—and this decision can have very important consequences for the player.
5. **River crossings.** In the original OREGON, there were no rivers and no crossings. I made river crossings a major feature of the new game. At each crossing, the player must make crucial decisions, based on the current conditions, which are strongly affected by the recent weather, which differs with each play of the game.
6. **A re-imagined hunting activity.** In the original OREGON, you hunt by typing the word “BANG.” Our goal was to turn the hunting activity into a bona fide skills-based arcade-style game, incorporating the actual terrain in which you are hunting, such as prairie, mountains, or desert. Likewise, the species of animals you find are based upon where on the trail you are currently located.
7. **A point system.** In the original OREGON there is no point system. In order to build a high level of replayability into the product, I designed a point system that is closely tied to the difficulty level you choose. Furthermore, as another motivator, the high-score list is pre-populated with a set of historical names covering a wide range of scores.
8. **Family members.** In the original OREGON, there is little mention that other people are traveling with you. To make the experience more personal, my design requires each player to provide four additional names as traveling companions. If you make bad decisions or encounter a streak of bad luck, then members of your family or group will die.
9. **Talking to people.** There is no conversation system in the original OREGON. In my new design I allow the player to talk to people along the way. This is an important method for obtaining helpful hints and discovering historical and geographic details of interest. Furthermore, it makes the game seem much more human.
10. **River rafting game.** The original OREGON does not mention rafting down the Columbia River, which I felt was a fascinating bit of history, as well as a great opportunity to include another arcade game in the product. Therefore we built the river rafting game, albeit a simpler version than what we had originally intended.
11. **Tombstones.** Another famous feature we added is the ability to create a tombstone for yourself after you die. This humorous interlude compensates for the player’s failure to complete the journey. But the real hook is that we store the tombstone data on the game disk, so that the next player who uses the same disk can see the tombstone.
12. **A detailed health model.** The original OREGON does not track the player’s health from turn to turn. My new design included an ongoing health model for the entire party, along with tracking individual illnesses and injuries. The weather, food rations, and pace all have cumulative effects on the party’s health, which can improve or decline over time.
13. **A detailed weather model.** The original OREGON did not include a true weather model, with seasons and data tied to several distinct climate zones. (The original had only two climate zones and no seasons.) In the new game, the program computes the weather every day, based on the current month and location. This in turn affects many other factors — such as health, river crossings, availability of water, and so on.
14. **Robust resource management.** My new design provided many new ways for a player to manage his resources, including: 1) adjust the pace of travel, 2) purchase specific types of spare parts, 3) rest at any time to improve the party’s health, 4) barter with other travelers to acquire essential resources, and 5) buy additional oxen at forts.
15. **A captivating travel screen.** The travel screen in the 1985 product—featuring a tiny animated ox and wagon—has become the iconic representation of the entire history of *The Oregon Trail* game. The combination of elements on this screen is both highly memorable and highly effective.
16. **Matt’s General Store.** In the original game, you never interacted with people. In the 1985 design, we introduced Matt’s General Store. Matt is the friendly store proprietor who assists you with your purchases, providing helpful advice as to what you should buy and how much you should buy.
17. **Specific diseases.** In the original game, if you get sick, then the disease is never named. (After you die, then pneumonia might be mentioned.) For the new design, I incorporated five diseases or symptoms often named by travelers on the Oregon Trail: typhoid, cholera, measles, dysentery, exhaustion, and fever. Oddly, one of these five has captured the popular imagination and become a huge meme. (“You have died of dysentery.”)
18. **Difficulty levels.** To encourage replay, I built three difficulty levels into the game. I chose three distinct professions to represent the three difficulty levels: banker, carpenter, and farmer. These choices have also become a meme.
19. **Trading with people.** At any time between landmarks, the player can attempt to swap items with passers-by. This can be quite helpful under certain circumstances, especially if a crucial wagon part has broken. However, this module is a pale shadow of the trading system I had intended to include. I later included the full, robust trading system in the game *Lewis & Clark Stayed Home*.
20. **Options to rest or change the pace.** The player can choose to speed up or slow down the pace, or to stop altogether for a few days of rest. These decisions represent a major tradeoff between the health of the party and progress on the trail.
21. **Period music.** At each landmark along the way, you can hear a different melody. Furthermore, every melody is an actual tune that was popular at the time of the Oregon Trail. Unfortunately, this innovation can be more annoying than helpful, especially on the Apple II, which has no volume control.

Taken together, these 21 innovations resulted in a very different experience compared to the earlier versions of the game. The new game was longer, with a much richer set of opportunities for decision making. It also allowed players to succeed in more than one way — by using various possible strategies and gameplay techniques. This was essential for my goal of making the game just as appealing to girls as it was to boys, as well as appealing to a broader range of tastes. In the end, the new game proved to be immensely popular with an exceedingly wide audience.

Of course, another big difference is that the original OREGON was a text-based game, while the 1985 design was full of color graphics. The decision to include graphics was an obvious necessity, not an innovation. On the other hand, some of our decisions about how to include the graphics were indeed innovative. For the 1985 game, we incorporated as many graphical details as we could fit on the Apple II disk, including the animated travel screen, animated river crossings, the hunting game, the rafting game, and a full-screen graphic for each of the landmarks.

Every now and then, I run across an online article that suggests that the 1985 Apple II version of *The Oregon Trail* was essentially identical to the original 1971 game, except for the addition of color graphics. This assertion is so ridiculously inaccurate that it completely baffles me. In fact, most of the features and details that people associate with the game were first invented for the 1985 product. On the other hand, my new design would never have been created if not for the great success of the original. Furthermore, the original version introduced seven key concepts upon which all later versions of the product were based. But after my 1985 design came out, every subsequent version treated the 21 major new concepts from that design as part of the essential core original design. Likewise, many of the lesser details from the 1985 design also become incorporated into later versions.

## Why the Process Succeeded

In the end, we met and exceeded all the goals for this project—except that the timeline and budget ran somewhat longer than we had anticipated. However, the product was completed in time for the autumn 1985 release to the home market, as intended. True to the goals, the product became a massive hit—not only in the home market, but also in the school market.

We certainly ran into some bumps and detours as we designed, built, and tested *The Oregon Trail*. And yet, overall, the process was quite successful. Looking back at the details of the project, I see several lessons regarding how to plan and start a new project:

**1. Seek lots of input, but don’t succumb to design-by-committee.**

I wanted all members of the product team to express their thoughts and contribute ideas. However, a committee will seldom produce a good design, nor can a committee make decisions quickly. Therefore, for the initial stages of design, I started with a divergent phase, when ideas are generated, followed by a convergent phase, when the ideas are narrowed down. Lots of people can and should participate in the divergent phase, but no more than three people should participate in the convergent phase for any particular module. After the winning concepts are selected, most of the design work still lies ahead, and most of that work will be done by individuals. But throughout the project, team members provide regular feedback to one another. And at key points in the project, feedback is solicited from people outside of the team, and user testing is conducted. In this way we continue the balance between getting ideas from many people, yet making decisions with very few people.

**2. Create the design in stages of successive refinement.**

Before attempting to design *The Oregon Trail*, I felt it important to define the high-level ideas, starting with the goals: Why are we creating this product? Who is the target audience? What will be our criteria for success? Then I proposed a general framework for the product, along with the key modules and features. After that, we defined each of the modules and features to a medium level of detail—enough to communicate how each would work. Finally, I fleshed out the full details of every module and feature. Meanwhile, as the product took form—first as prototypes and then as actual code modules — I continued to make design improvements. Each screen went through successive waves of scrutiny to improve the layout, text, and usability, and the underlying mathematical equations went through rounds of tuning to improve the gameplay. I doubt that it would have been possible to design everything first, in a single huge document, before starting work on building the product. Attempting to create one huge document at the outset would have been wasteful at best—and catastrophic at worst.

**3. Create prototypes.**

While it is helpful to draw sketches of the proposed screens, a prototype provides additional information. The first objective of prototyping is “visualization”—allowing people to see a vision of the finished product, even if only in rough form. As a second stage, the prototype can be made partially interactive, allowing people to get a taste of playing the finished game. Both of these stages provide useful information for adjusting the design relatively early in the project—certainly much earlier than if we had waited for a beta version before showing it to anyone.

**4. Test the prototypes and early draft software with users.**

It is certain helpful when teammates and other colleagues evaluate the prototypes—but this alone is not adequate. It is also crucial to investigate how real users react to the product—as soon in the development cycle as possible. No matter how experienced and insightful the designers are, there will always be surprises when real users try out the product. These surprises range from small details to much larger issues. For *The Oregon Trail*, one of the small surprises was that the kids did not understand the word “outfits”—which I then changed to “sets of clothing”. A medium-level surprise was that many players were confused by the screens where you name the members of the party, so I revised those screens. And the most significant issue we found was that the travel screen in the alpha testing did not hold the interest of the players. And so we redesigned this screen, which then became the most famous screen in the product.

**5. Constantly prioritize what needs to be done next.**

On nearly every project I have worked on, we had to scale back the plans at some point, either subtly or dramatically. This is a key reason to prioritize the various components. Always put an early emphasis on the modules and features that are most likely to be in the finished product.

Risks are another factor to consider. Any project has risks, which are tied to unknowns. To reduce the risks, it is crucial to explore the unknowns early in the project. Some risks are technology based—Will this module be fast enough? Can we get everything to fit on the disk? Some risks are user based—Will users be able to complete this complicated task successfully? Will users understand the meaning and details of this crucial screen? The priorities early in the project are to design and build prototypes or proof-of-concept modules that address the principal unknowns, thereby increasing the odds of success.

This is exactly what we did with *The Oregon Trail*. In the earliest stages we decided what was most important to illustrate with the internal prototype. Then we decided what features were most important to test with kids, and we included those in the Alpha version. Immediately after that, we prioritized all the modules and features that were planned for the product, on the assumption that not everything would make it into the finished product. From a fairly early stage, we also drew up technology plans and conducted technology tests to address the technical risks.

## Summary and Thanks

In the course of my career, I have worked on countless software projects. For the vast majority of these projects, I have every reason to be proud of the results—while also being aware of certain details that could have been further improved. However, except for *The Oregon Trail*—and to a lesser degree *Number Munchers*—none of these projects resulted in a product that became truly famous—even though some of them (especially the web-based applications that I designed for corporations) did indeed get a lot of use.

The upshot is that I feel thankful for whatever fame my products have been able to achieve. I am particularly thankful for having had the opportunity to work on *The Oregon Trail*—and for the combination of circumstances that allowed the product to become such an outstanding success. I am thankful for having been able to work with highly talented and dedicated colleagues—whose collaboration played a huge role in the successful outcome. (My principal collaborators were Charolyn Kapplinger, John Krenz, Shirley Keran, Bob Granvin, and Roger Shimada.) I am thankful to MECC for assigning me to this project as the lead designer and project leader—and for giving me the latitude to steer the project as I saw fit (most of the time). I am thankful that Don Rawitsch, Bill Heinemann, and Paul Dillenberger invented the first brilliant version of the game, which spawned all subsequent versions. And I am thankful that millions of people have been able to play and enjoy this game—and that so many of them fondly remember the game after all of these years.

*R. Philip Bouchard spent 32 years designing computer software, 18 of which were focused on educational software. He was the principal designer for the Apple II games* The Oregon Trail *and* Number Munchers*. You can read more about his design experiences at his* [*blog*](https://medium.com/@philipbouchard)*, where this article was originally published.*