

# The Making of King's Quest IV

Spring 1989 (Sierra Newsletter)

Donald B. Trivette

## - The Story of How a Video Animated Adventure Game Is Made -

"King's Quest IV," like the other Quests in the saga, began as an idea in Roberta Williams' head. This time she wanted the main character to be a woman, so Princess Rosella, the sister of Gwydion from "King's Quest III," was given the title role. Rosella's quest is to find a magic fruit that will restore her father, King Graham of "King's Quest I" and "King's Quest II," to health.

First Roberta outlines the story on scraps of paper and in notebooks, making changes freely as the plot thickens. Sometimes she consults her two children, D.J. and Chris, but husband Ken is little help; he's more interested in the technical details than the story line. As the story begins to gel, Roberta makes rough drawings of each scene, called "rooms" by the developers. For example, the original sketch of the fisherman's shack, room 7, showed a boxy house with a line extending out to represent the pier.

Roberta also writes a description of each room. Here's what she says about room 7:



A beach. Ocean to the west. Beach turning to green grass to the east. A fisherman's shack overlooks the ocean with maybe a little pier or something going into the water.

In addition to the drawing and the description, Roberta creates the script or story line that explains what goes on in each room. Although this gives away a few clues, here's what she says about the fisherman's shack:

**1.** You need to go to room 7 and enter the fisherman's house, room 42. There will be a fishing pole in there leaning against the wall. You want to get the fishing pole but to do that you need to buy it. The fisherman and his wife will take the pouch of diamonds in trade for the fishing pole. But, they will also take the golden ball. You don't want to give them that, though, because you need it for the frog. It's not possible to give them the hen that lays the golden eggs because you can't get the hen until AFTER getting the unicorn.

**2.** Next, you can go into any of the beach rooms and "fish." Randomly you can catch a fish. Keep the fish with you. (Maybe you can only fish from the pier, for animation reasons.)

Of course some things change as the game develops. To conserve memory, it was decided that you could only fish from the end of the pier. And to avoid confusion, the fisherman and his wife will take nothing but diamonds in exchange for the fishing pole.

## -- Setting the Stage --

From Roberta's sketches and written material, the artists and programmers begin to construct the game using special programming tools developed by Sierra. View Editor is the program used to draw animated objects like Rosella and the fisherman. The central character, Rosella in this game, is always called "Ego" by the programmers and artists no matter what his or her gender in the program.

Picture Editor is the program used to construct and draw the background scenes like the fisherman's shack and pier. When the background artist got Roberta's sketch of the fisherman's shack, he first drew an outline of the building and pier. Next, he colored in large surfaces, then added textures and details like flowers, roof shingles, and wood grains. All "King's Quest" rooms use shade and shadow as well as perspective to give the illusion of three dimensions. Vanishing points are chosen to be high on the horizon to give the characters as much space as possible to walk; a low vanishing point would limit Ego's range.



Backgrounds are not stored as a completed picture; instead, they're constructed and stored as coordinates and vectors. Vectors give the instructions for drawing a picture, and they have the advantage of taking less space than would a bit image of a complete picture. The artist works with a standard IBM PC and mouse, using pull-down menus and windows similar to many commercial CAD programs.

To maintain and enhance the three dimensional quality of the rooms, every object is constructed with a priority in relation to the other objects. There are 16 bands or areas in which things may be placed. Although the priority bands are invisible in the finished product, the artist must use them like a horizontal grid as he draws the room. Considerable effort and time is spent placing houses, bushes, and trees so the player remains unaware of the room's mathematical rigidity. It wouldn't do for things to look like they were lined up on a checker board.

As the room nears completion the artist adds control lines that determine where Ego can walk. She shouldn't walk through a wall or tree, for example, but sometimes she does. If a tree is placed between two priority bands, Ego will walk right through it. The Sierra staff spends a lot of time running Ego all over the rooms looking for places where she falls off or walks through something she shouldn't. Fixing an error may involve shifting a tree to a priority bank or adding new features to a room to cover up a

programming bug. In the fisherman's shack room, for example, a rope fence was added to keep Ego from falling off the pier onto the sandy beach.

## -- Creating the Characters --

Another artist is responsible for drawing the animated characters. Although the backgrounds are vector graphics (lines), the characters like Ego and the fisherman are made up of pixel drawings. The fisherman, for example, is created in a box about 33 squares high and 14 to 18 squares wide. By changing the colors of the squares, the character takes on different shapes, activities, and appears to move. The background color is special. This invisible color takes on whatever color is behind it in the room. Thus, as Ego walks by a tree, you see brown bark surrounding her arm. The invisible color varies from room to room, but it's always a hue different from everything else in the room.

Animation is achieved by rapidly displaying several of these drawings or cels one after the other, similar to a flip book. It takes eight cels for the fisherman to take one step to the right, and eight more cels for one step to the left. A step up or a step down requires a cycle of six cels each. Thus, 38 separate drawings are required for the fisherman to walk. Because he sometimes carries a fishing pole, another 28 drawings are needed. He needs six cels to eat, six more to talk, and eight cels to get up off the pier. In all, the fisherman needs 76 drawings for his animation. Ego herself requires more than one thousand drawings to animate all her activities.



A room may take several days to draw, and then additional time is spent debugging and cleaning up. If the room takes too much memory (the average is about 12,000 to 14,000 bytes) it will load slowly and waste space. As part of the clean up, the artist looks for places to economize. The fisherman's shack takes 12,700 bytes and, as this is written, there is debate about the best way to draw the waves. Continuous waves will take considerable memory, so when you get to this room you may find the roaring surf reduced to an occasional swell.

## -- Sierra Creative Interpreter --

The programmer is the one who puts all the rooms and animated characters together to make the game run. He does this using a special computer language called SCI (Sierra Creative Interpreter) developed by Sierra. SCI is an object-oriented language similar to LISP; it's written in Assembler, C, and itself. An

older version of the interpreter, called Adventure Game Interpreter (AGI), was used to create "King's Quests I," "II," and "III."

With SCI the programmer established classes of objects. The Actor Class, for example, includes anything that moves such as sea gulls, ogres, and Ego. The Props Class is composed of items that move but don't go anywhere -- like a candle flame or smoke. The Views Class is for things that don't require any animation, like the fisherman's fishing rod. The programmer writes scripts in the SCI language that define how the objects come together and interact with one another.

In room 7, a fairly simple room by "King's Quest IV" standards, there are six things to animate: smoke, sea gulls, waves, the door, the fisherman, and Ego.

A repeating cycle of four drawings gives the illusion of smoke coming from the fisherman's chimney. The word = 3 statement causes the smoke to move at one-third full speed. The instructions to make the smoke are:

```
animate.obj( smoke);
ignore.horizon ( smoke);
set.view( smoke,v.fish.cabin);
set.loop( smoke, 1);
position( smoke, 95, 16);
work=3;
step.time( smoke, work);
cycle.time( smoke,work);
draw( smoke);`
```

Opening and closing the door to the house is considerably more complicated. This requires activating a five-drawing sequence which is triggered by the following code:

```
if (said( open,door)) [must be close enough]
{if (posn( ego, 86, 120, 106, 133))
{if (!night)
{if ( door,open)
{print("The door is already open.)}
else {set( game.control);
set.priority( ego, 11);
start.update( door);
end.of.loop( door, door.done); }}
else {print("You can't -- it's locked.}}
else{set(notCloseEnough); }}
if(said( unlock, door)) [ must be close enough]
{if (posn( ego, 86, 120, 106, 133))
{if(!night)
{print("The door is already unlocked...}
```

```

else {print("You can't, it's locked.  }}
else{ set ( not CloseEnough);
if((said ( knock, at, door) // [ must be close enough
said( knock) //
said( knock, on, door) //
said(knock, door)))
{if(posn( ego, 86, 120, 106, 133))
{if(!night)
{print("You knock on the door...
a woman says...  )
else {print("You knock on the...
a man calls out...}}
else (set( notCloseEnough);  }}

```

In English this says: If you try to unlock the door the program will say the door is already unlocked. If you knock on the door -- the polite way to enter a home -- the fisherman's wife will invite you in and you can open the door. At that point the program will begin the animation to open the door. If you're not within range of the door, the program will tell you that you're not near enough. If you should come to the house at night, a whole different set of instructions apply (the fisherman will say it's too late).

The sea gulls require a good bit of code to fly about in room 7. Their number (from 1 to 3) is randomly determined and their actions are governed by a built-in Wander function. The sea gull's bounds are limited by control lines and priorities. There's a control line just about the horizon, so the birds won't wander into the ground, and their priority is set to 0 so they will always appear to fly behind the house.

A function is a set of generic instructions that can be used over and over to do the same operation on different sets of data. The BASIC language, for example, has a square root function that calculates the square root of a number; spreadsheets have an average function to find the average of a column of numbers; and SCI has many functions to speed up programming, including Wander, Chase, and Moveto.

Chase is a tricky bit of programming; it takes a lot of instructions to assure that Ego gets chased properly. At all times the Chase function knows her coordinates as well as those of the chaser. When Ego moves around a rock or tree, the Chase function must compute a path for the chaser -- perhaps an Ogre or Wizard -- to follow.

If Ego gets caught, there is no Death function to kick in and automatically take care of the details. Each death scene gets individual animation and attention; there's no standard way for her to die. Incidentally, Ego can't die in room 7 (there's no animation for that to happen); if she falls into the water, she will automatically swim.

There is animation in room 7 in case she falls off the pier. In fact, there are several scripts for that event. One if she falls off the north side and another if she falls off the south side. If she falls south,

Score: 33 of 230 KQIV The Perils of Rosella



for example, that triggers instructions called EgoShortFall and EgoShortSplash. A similar code provides for a fall to the north, but without the splash loop since a splash would be hidden by the pier.

There is no script if Ego should happen to fall off the pier onto the beach, so the artists added the rope fence to prevent that from happening. Think how silly she would look splashing around on dry land.

Ego can also fall on the rocks -- called the

fallRocks Script; this triggers an EgoStunned routine for three seconds.

Control is taken away from the user as the fall begins and is returned when the fall sequence is complete. In other words, you can't use the joystick or keyboard while Ego is falling or stunned.

Room 7 is an average "King's Quest IV" room, yet it takes more than 600 lines of instructions to describe how things happen there. A programmer spent two days writing room 7 code, and the cleanup programmer spent more time improving and correcting the code. "King's Quest IV" distinguishes between day and night, so there are about 30 rooms that have moonlight versions of the daylight rooms. The entire "King's Quest IV" game contains more than 140 rooms and tens of thousands of lines of instructions.

## -- Music --

Although "King's Quest I-III" have music and sound effects, the quality of the music hadn't kept pace with the improvements in graphics and animation. Roberta was determined to change that, so in addition to artists and programmers, "King's Quest IV" also had the talents of a professional musician. Once the game was playable, a tour of the quest was recorded on videotape and went to William Goldstein for musical scoring.

Goldstein has several Hollywood films to his credit including Touchstone's recent "Hello Again," as well as the television version of "Fame," for which he received an Emmy nomination. His electronic music compositions include "Ocean Scape" on the CBS label and "Switched on Classics" for the Proarte label.

For "King's Quest IV" Goldstein chose to work with a Roland MT-32 synthesizer, which is actually eight synthesizers in one. With its capability to simulate 32 voices or instruments at the same time, the Roland MT-32 is like having a small orchestra in your computer. The stereo synthesizer can add rich and subtle ambiances to a musical composition. Goldstein used a slight echo in each of the interior rooms, for example, to give an illusion of depth and space.

There are more than 75 pieces of music in "King's Quest IV," which is more than in a feature-length film. The compositions include a sinister wedding march, organ music, Middle Eastern music to charm a cobra, and five selections for a wandering minstrel, who plays so badly that you'll laugh at his rendition of "Greensleeves." There's an elaborate eight and a half-minute piece to accompany the animated introduction where the old King is taken ill and Rosella learns of her quest. This is mood music to set the tone and emotion for "King's Quest IV." In addition, each of the 35 characters in the game has his or her own theme songs.

Although room 7 doesn't have any music, the fisherman has an introductory tune when he meets Rosella on the end of the pier. Roberta Williams suggested several adjectives that describe the fisherman's character: "old, sea salt, poor, despairing" and Goldstein composed a suitably salty eight-second chantey.

To get the full stereophonic, orchestral effect of the music, you'll have to install an optional music card in your computer. For IBM computers, "King's Quest IV" supports three cards: The AdLib music card, the IBM music card, and the Roland MT-32. They range in price from \$245 to \$600. The music cards must be connected to the amplifier and the speakers in your existing stereo system.

For computers without enhanced music capabilities, Goldstein's soundtrack was distilled into a form that uses the limited capability of the standard IBM PC. The Apple and Tandy versions of "King's Quest IV" have better sound than the standard IBM version, because those machines have better sound hardware.

The striking composition of William Goldstein and the "live-performance" sound of a music card contribute a new dimension to "King's Quest IV." You've got to hear it to appreciate how much of a difference professional music makes in a computer game.

### **-- Quality Control --**

Once the game is thought to be finished, it goes to the Sierra Quality Assurance Department along with the story line and the programmers' maps. One of the first things QA does is to extract from the software all the text and messages -- things like, YOU CAN'T DO THAT and ROSELLA SEES A BEAUTIFUL FAIRY... This results in a small book, which is then proofread for grammar, spelling, and punctuation by one of the five quality assurance people.

The QA staff begins playing the game using the programmers' maps and notes. Even with this help, it takes at least two eight-hour days for them to complete the quest. The testers are looking for inconsistencies: places where Ego is visible through a tree, where she can't leave a room, where she is carrying the wrong object. Errors are reported to the programmers, who decide whether they are valid errors. In "King's Quest III," for example, when the wizard demands a meal, Ego must leave the dining room and then re-enter in order for the wizard to sit at the table and eat. The QA people thought this was an error, the programmers said it wasn't. So if you put food on the table, but the Wiz won't sit

down, try leaving and returning. The QA staff also makes esthetic suggestions. They felt the antlers in the dining room in the Apple version of "King's Quest III" were the wrong color. The color was changed. Another color problem in the same game was not so easily resolved. There is a spider that gets carried out over the ocean and dropped into the water (if he doesn't get Ego first). The spider and the ocean were the same color. The QA people, the programmers, and the artists had a meeting to discuss the problem of the invisible spider. Unfortunately, all the alternate colors were also in use and just caused the spider to disappear somewhere else. This colorful matter was bucked to the top: Roberta decided the spider and the ocean would remain the same color and the spider's splash was enhanced.

The Quality Assurance Department uses more than 30 different computers in the most popular configurations to test each Sierra product. About half of the machines are MS-DOS clones and compatibles; the others are Macintoshes, Apples, Amigas and Ataris. After four to eight weeks, hundreds of man and woman hours, and 20 to 30 error-correcting cycles, the product is deemed ready to ship. Even with all this careful testing, the QA staff knows a few of us will try something or do something the designers and testers did not anticipate. There are usually a small number of follow-up bugs to exterminate...