Work the Shell

End Game

Dave Taylor

Abstract

Want to bet you can find a problem in our final Blackjack script?

This is the last column in our Blackjack series, and in this column, I show the final snippets of code needed to weave all of the disparate pieces of the game script into a playable game. For obvious reasons, I can't present the entire script here in the magazine (it's almost 300 lines long), so instead I highly encourage you to pop over to the Linux Journal FTP site and grab a copy of the script as you read along (ftp://ftp.ssc.com/pub/lj/listings/issue148/9051.tgz).

As with many betting games, Blackjack has evolved to have many esoteric rules with splitting pairs, insurance and various other things that take something relatively simple and make it more complex. We'll ignore all of that, however, and also ignore the betting component of the game too (this is a [geek] family magazine, after all) and just focus on the game play.

Dealer Gets One Card Up

Therefore, the first thing we need to know is that the player can see both cards as dealt and one of the two cards that the dealer deals for itself. That's the first piece of code we need to add, and because we aren't allowing insurance or betting, it needs to be included immediately after the tests for blackjack in the code:

```bash
echo -n "** Dealer's hand: "
showCard ${dealer[1]} ; echo -n "$cardname, "
echo "(card face down)"
echo ""
```

From a strategic perspective, if you have even a rudimentary grasp of probability, you'll know that cards with a value of ten are far more likely than any other card value in the deck. If the dealer has an eight or nine showing, for example, odds are very good that it has 18 or 19 as a hand value. That'll, therefore, change your own playing strategy too, perhaps making it more likely that you'd take another card if you have an interim hand value of 17; whereas, if the dealer had a five showing, for example, you might be more likely to stick with a hand value of 16.

In previous columns, you've already seen the basic script to display and calculate a hand's value, so this is nothing new:

```bash
echo -n "** Dealer's hand: "
showCard ${dealer[1]} ; echo -n "$cardname, "
showCard ${dealer[2]} ; echo "$cardname"
```
handValue ${dealer[1]} ${dealer[2]}

But, now we're going to add a loop below this that will keep taking cards until the hand value is 17 or higher (that's standard Las Vegas Blackjack rules: 16 and lower the dealer “hits” or takes another card, and 17 or higher the dealer “stands” or sticks with its hand).

This is a bit tricky, so take your time reading it:

```
while [ $handvalue -lt 17 ]
do
    dealer[$nextdealercard]=${newdeck[$nextcard]}
    showCard ${dealer[$nextdealercard]}
    nextcard=$(( $nextcard + 1 ))
    nextdealercard=$(( $nextdealercard + 1 ))
    echo "" ; echo "** Dealer takes: $cardname"
    handValue ${dealer[1]} ${dealer[2]} ${dealer[3]} \\
    ${dealer[4]} ${dealer[5]}
done
```

With some good routines and variables already in place, it turns out to be surprisingly succinct to have the dealer play its hand out. Hurray for that bit of good design!

**Who Won the Game?**

Now that we have the game-play logic, it's simply a matter of having the set of conditionals to figure out who won or whether the game ended with a tie:

```
if [ $handvalue -gt 21 ]
then
    echo "**** Dealer busted! Player wins with \ 
    $playerhandvalue!"
    playerwin=$(( $playerwin + 1 ))
eelif [ $handvalue -eq $playerhandvalue ]
then
    echo "**** Dealer and player tie with \ 
    $handvalue points."
eelif [ $handvalue -lt $playerhandvalue ]
then
    echo "**** Player wins with $playerhandvalue"
    playerwin=$(( $playerwin + 1 ))
else
    echo "**** Dealer wins with $handvalue"
    dealerwin=$(( $dealerwin + 1 ))
fi
```

Whaddya think? Enough code listings?

Let's see how the game plays now:

```
$ sh blackjack.sh
**** Welcome to Blackjack.sh! ****
Dealer's hand: Queen of Hearts, (card face down)
```
You've been dealt: 2 of Spades, 7 of Hearts

H)it or S)tand? (recommended: hit) hit

You've been dealt: 10 of Diamonds

H)it or S)tand? (recommended: stand) stand

You stand with a hand value of 19

Dealer's hand: Queen of Hearts, 4 of Hearts

Dealer takes: Queen of Clubs

**** Dealer busted! Player wins with 19!

You can see that we had 2S and 7H (9 points), took a card, 10D, giving us 19 points. We stayed with that, and the dealer revealed that it had QH 4H, 14 points, and took another card which proved to be QC, which took the dealer over 21 points. We won!

There are a few more nuances to the program, including keeping track of how many times the dealer or player wins, and at the point where you're asked “hit or stand”, you now can type in quit (or q) to quit. Then, it'll show you:

H)it or S)tand? (recommended: stand) q

Player quits. Standings: dealer wins: 5 and player wins: 3

There's Still a Bug

There's also one outstanding bug in the code, and I invite you to dig around and figure it out. If the dealer is dealt an Ace in play, it's automatically counted as 11 points, not one or 11, as it should be. Your challenge: figure out where that problem arises and how to fix it. If you think you know, e-mail me your proposed solution (perhaps a diff of your version versus the version on the Linux Journal FTP site), and we'll see how you did.

Next month, we'll crack open a completely different shell script task and see what we can do to help your day-to-day Linux administrative tasks, because we've probably wasted plenty enough ink on a casino game for this year! See ya then.