Part 4 Requirements

Design

Dr. Mark Dogfurry calls you in for an emergency meeting.

Dr. Dogfurry: *"We have a crisis. Some players are irate that they sometimes receive no prize when they have participated in a successful quest. For discussion, let's call a piece of loot rewarded to a player in a quest a 'prize'."*

You: "I think I have an easy solution. Why don't we add enough additional prizes (loot) retroactively — to past — quests so that we can assign the additional prizes to those players who did not receive anything?"

"Yes, that is what I first thought too, but Corporate is refusing to allow us to add any more prizes. They think that virtual prizes are actually worth real money! So unfortunately, that won't work."

"Instead, let's do the following. We'll take prizes from players who received more than one prize in a quest, and reassign those prizes to players who got none. That way, everyone will have gotten something. I call this 'Project Robin Hood'. We take from the rich players and give to the poor players! It might make some players mad to have prizes taken away from them … but what can you do?"

"Will that always work? What if there are too many players in a given quest who got nothing to redistribute so everyone has at least one prize?"
"Hmm. You're right. Well, we may not always be able to give everyone a prize in every case. But we can come as close as possible. I suggest the following strategy."

1 *“For each quest, identify an actor — a player who participated in the quest — who has more than one prize (if there is any). Call this actor the ‘donor’.”*

2 *“For each quest with a donor, pick one of the prizes he or she had received. Call this prize the ‘donation’.”*

3 *“For each quest, pick a player who received nothing (if there is any). Call this player the ‘recipient’.”*

4 *“Finally, for each quest with both a donor and a recipient, reassign the donation prize from the donor player to the recipient player!”*

“That … sounds complicated to implement!”

“That's why we pay you the big money, a crack database programmer!”

“Oh, and that strategy even will not always work. A quest might have more than one potential donor and recipient. Your strategy will fix things for just one recipient per quest.”

“Hmm, true. After we do 1–4, there still might be donors and recipients in quests… Ah, I have a solution. We just repeat 1–4 until no prize-swap possibilities remain! Then we will have fixed things as best we can.”

“Yes, I think you're right. That would work. But how to do all this?! There are lots of details!”

“First, who should we pick as the donor? There might be more than one player per quest who could be the donor!”

“Let's do this. We pick the player who got the most prizes from the quest. If there is a tie, well, we'll chose the one whose prizes add up to the most scrip (sql) in the tie. Oh, and if there is still a tie, we'll just choose the ‘login’ of the ones in the tie at the bottom of the list. (Okay this last one is not so fair, but should rarely occur.) That's pretty easy to implement … in fact, this query ought to be what I mean: Q-Donor.”

“And for which prize of the donor — the donation — that we should take away … well, let's pick the one that is worth the least scrip. In case of ties, let's pick the one in the tie for which the treasure name is at the bottom of the list. Hmm … there still could sometimes remain a tie, because a quest can offer the same prize multiple times (distinguished by ‘loot#’). Well, in case of still a tie, we pick the highest loot# in the tie. Let's see, we can extend our earlier query for this: Q-Donation.”

You breathe a big sigh of relief after reading the queries. “Yes, I see. That looks right.”

“And to find per quest someone who did not receive a prize, that's easy enough! Ah … in case of more than one per quest, well, let's just choose the one with the login at the top of the list. This should do it: Q-Recipient.”

“Oh, wait! We might have a problem. What if we take a prize from a donor and give it to the recipient … then the donor might have no prize, so becomes a potential recipient. And the recipient is now a potential donor. When we loop, we might just swap prizes back and forth forever! Our procedure would not terminate.”

You think carefully about it. “No, we're safe. There are three classes of people per quest: *donors*, who have two or more prizes; ‘*singletons*’, who have exactly one prize; and ‘*recipients*’, who have no prize. Our procedure can turn donors into singletons, and recipients into singletons. But it never turns singeltons into anything else. So donors cannot become recipients, nor vice versa. Therefore, our procedure will terminate!”

“Whew! Good reasoning! Well, off with you then. What you'll need to do is join Q-Donation and Q-Recipient. Only for quests that have both a donor and a recipient can you reassign the donation prize from the donor actor to the recipient actor. That would be an update to the Loot table on ‘login’, setting the loot tuple for the donation prize to the recipient (away from the donor). And then just loop, until no possible swaps remain.”

“Good! Off with you to implement Project Robin Hood!”

1. <u>Create new app ‘steal’</u>