

# Strictly X-Factor

TV talent shows like *X-Factor*, or *Soapstar Superstars* have always been popular. Its not just the talent on show that make them must see TV – it's having the right mix of personalities in the judges too. Simon Cowell has made a career of being rude – even reaching the dizzy heights of a guest appearance on *The Simpsons*. In contrast judge Sharon Osborne's on screen persona is far more supportive. It's often the tension between the judges that makes good TV.

However, if you believe Dr Who, the future of game shows will be robot judges like AnneDroid in the space age version of *The Weakest Link*...let's look at the robot future. How might you go about designing computer judges?

We need to write a program. We don't want to have to describe new judges from scratch each time. We want to do as little as possible to describe each new one.

## **What makes a judge**

First let's describe a basic judge. We will create a plan, a bit like an architect's plan of a building. It can then be used to build individual judges. What's the X-factor that makes a judge a judge? First we need to decide on some characteristics of judges. We can make a list of them. The only thing common to all judges is they have different personalities and they make judgements on people. Let's simply say a judge's personality can be either supportive or rude, and their judgements are just marks out of 10 for whoever they are watching.

Character : SUPPORTIVE OR RUDE.  
Judgement : 1 TO 10.

So let's start to specify (describe) Judges as people with a personality and capable of thinking of a mark.

DESCRIPTION OF A Judge:  
Character personality.  
Judgement mark.

All we are saying here is whenever we create a Judge it will have a personal character (it will be either RUDE or SUPPORTIVE). For any given judge we will refer to their character as "personality". It will also have a current judgement, which we will refer to as mark: a number between 1 and 10.

## **Best Behaviour**

We are now able to say whether a judge is rude or supportive, but we haven't actually said what that means. We need to set out the behaviours associated with being rude and supportive. To keep it simple, let us say that the personality shows in the things they say. A rude judge will say "You're a disgrace" unless they are awarding a mark above 8/10. For high marks they will grudgingly say "You were ok I suppose".

TO Speak:  
IF (personality IS Rude) AND  
(mark <= 8)  
THEN SAY "You're a disgrace".  
IF (personality IS Rude) AND  
(mark > 8)  
THEN SAY "You were ok I suppose".

It would be easy for us to give them lots more things to choose to say in a similar way. We can do the same for a supportive judge. They will say "You were stunning" if they award more than 5 out of 10 and otherwise say "You tried hard".

## **Ten out of Ten**

The other thing that judges do is actually come up with their judgement. We will assume, to keep it simple here, that they just think of a random number – essentially throw a 10 sided dice under the desk with numbers 1-10 on. Judges' decisions can sometimes look like that on TV!

TO MakeJudgement:  
mark = RANDOM (1 TO 10).

Putting that all together to make our full judge description we get:

## **Our final plan for making judges**

DESCRIPTION OF A Judge:  
Character personality.  
Judgement mark.

TO Speak:  
IF (personality IS Rude) AND  
(mark <= 8)  
THEN SAY "You're a disgrace".  
  
IF (personality IS Rude) AND  
(mark > 8)  
THEN SAY "You were ok I suppose".  
  
IF (personality IS Supportive) AND  
(mark > 5)  
THEN SAY "You were stunning".  
  
IF (personality IS Supportive) AND  
(mark <= 5)  
THEN SAY "You tried hard".  
  
TO MakeJudgement:  
mark = RANDOM (1 TO 10).



### **Kind words for our contestants?**

Suppose now we want to create a rude judge, called SimonCoward. We can use the plan. We need to say what its personality is (Judges just think of a mark when they actually see an act so we don't have to give a mark now.)



SimonCoward IS A NEW Judge WITH personality Rude.

This creates a new judge called SimonCoward and makes it Rude. We could similarly create a rude AnneDroid:

AnneDroid IS A NEW Judge WITH personality Rude.

For a supportive judge that we decide to call SharONN we would just say:

SharONN IS A NEW Judge WITH personality Supportive.

Whereas in the specification we are describing a plan to use to create a Judge, here we are actually using that plan and making different Judges. So this way we can quickly and easily make new judge clones without copying out all the description again.

### **A classless society?**

Computer Scientists are lazy beings – if they can find a way to do something that involves

less work, they do it, allowing them to stay in bed longer. The idea we have been using to save work here is just that of describing classes of things and their properties and behaviour. Scientists do that a lot:

Birds have feathers (a property) and lay eggs (a behaviour).

Spiders have eight legs (a property) and make silk (a behaviour)

We can say something is a particular instance of a class of thing and that tells us a lot about it without having to spell it all out each time (even for fictional ones): eg Hedwig is a bird. (so feathers and eggs) Charlotte is a spider. (so legs and silk) So we can now create judges to our hearts content, fixing their personalities and putting the words in their mouths based on our single description of what a Judge is.

### **All Change**

We have specified what it means to be a robotic judge and we've only had to specify the basics of Judgeness once to do it. That means that if we decide to change anything in the basic judge (like giving them a better way to come up with a mark than randomly or having them choose things to say from a big database of supportive or rude comments) changing it in the plan will apply to all the judges of whatever kind.

What we have created is our first object-based program – it would be relatively easy to convert this into a program in a programming language like Java or C#.

We could create robot performers in a similar way (after all don't all the winners seem to merge into one in the end?). We would then also have to write some instructions about how to work out who won – does the audience have a vote? When do judges make judgements? When can they speak their mind? How many get knocked out each week? That's no harder. Why not give it a try and judge for yourself?

So how does a SharONN Judge have daughter KeLEE judges without any help from Ozzie? See [www.dcs.qmul.ac.uk/cs4fn/](http://www.dcs.qmul.ac.uk/cs4fn/)

