PCGToy: a randomization tool for GMs  
Ian Horswill, ian@northwestern.edu

# Introduction

PCGToy is a simple tool for generating game entities like enemies, NPCs, magical items, quests, and the like. It allows you to specify a class of entities in terms of its attributes, called **variables**, and their possible values, called their **domains**. You can add variables and domain, populate the domains with possible choices, and then press the **Roll button** to make a set of random choices.

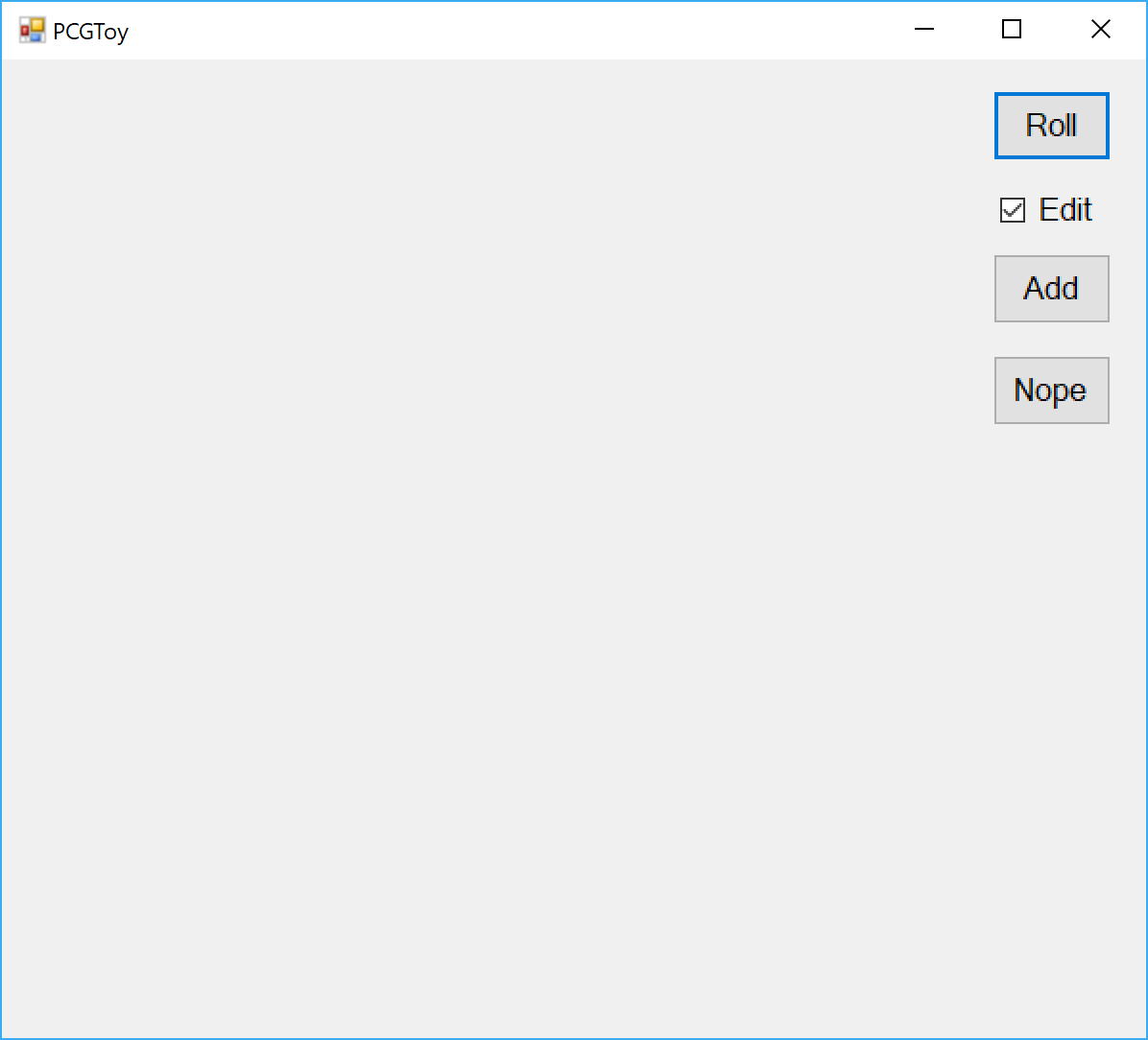
What’s unusual about PCGToy is it allows you to specify constraints on the variables. You can do this in two ways:

* You can make a variable conditional, meaning it only appears when some other variable has a specified value. For example, NPCs might have a caste attribute, but only when they come from a particular nation or race.
* You can outlaw particular combinations of attribute values. For example, you might stipulate that thief characters can’t be lawful by designating the combination class=thief, alignment=lawful as a **nogood** combination.

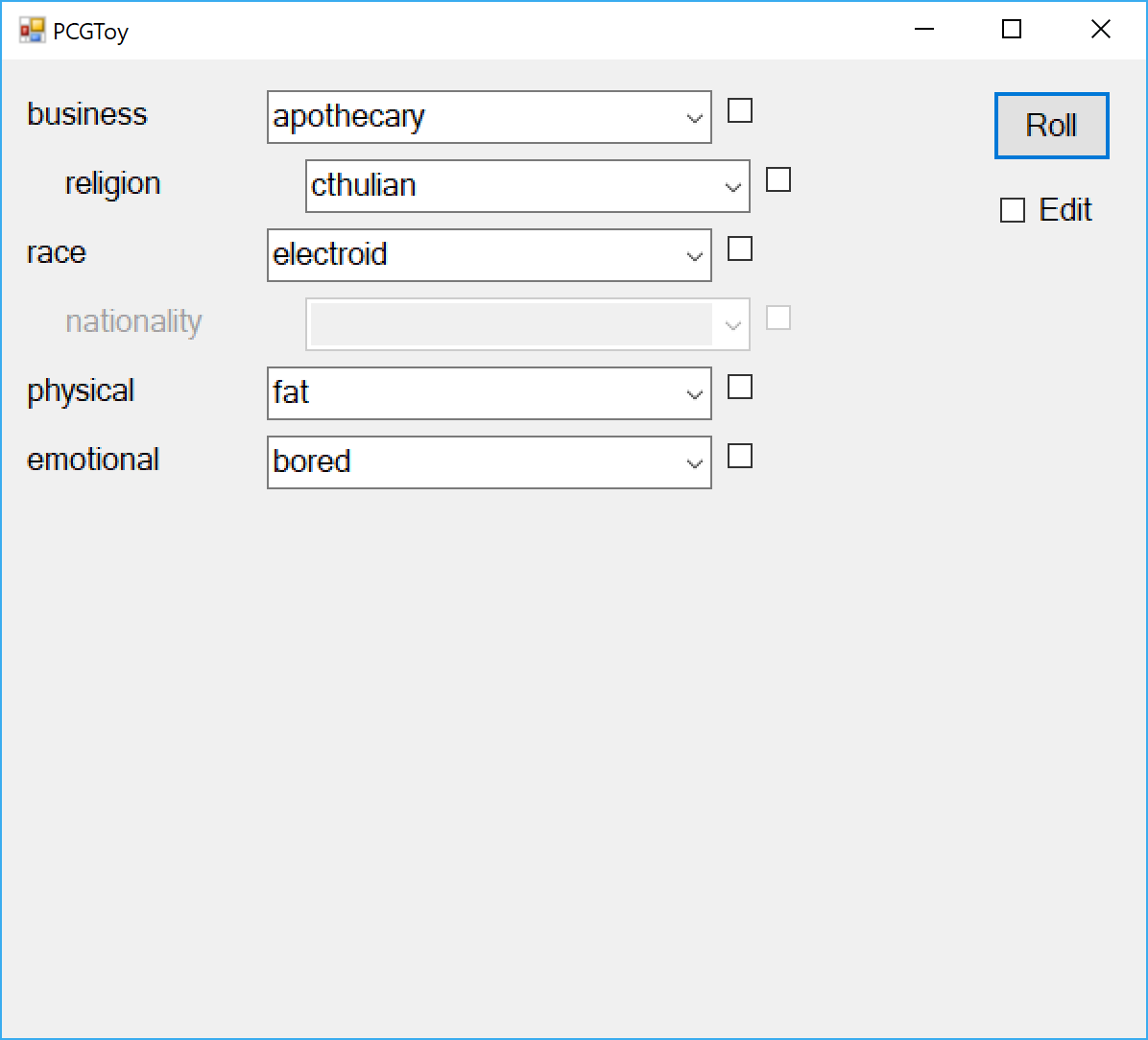
You can incrementally add new variables, values, and nogoods as you like, using the tool, and then save them to a file.

# Walkthrough

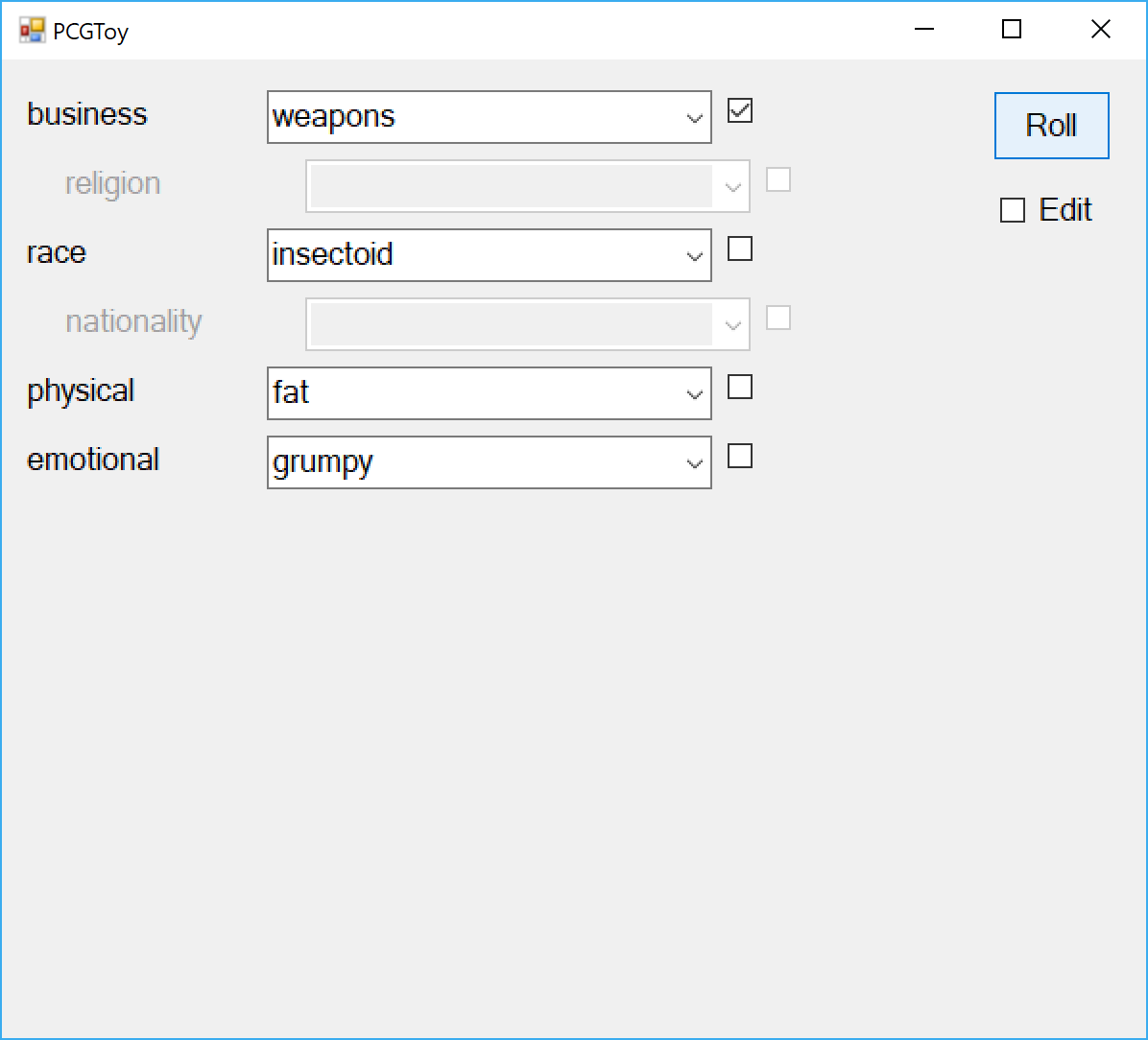
Run PCGToy. You should see something like this:



This is showing you a generator with no variables defined, and a set of buttons for letting you add variables. Before we do that, let’s use a pre-built generator. Type **Control-O** to open a file. Have it open **merchant.scm**. You should see something like this:



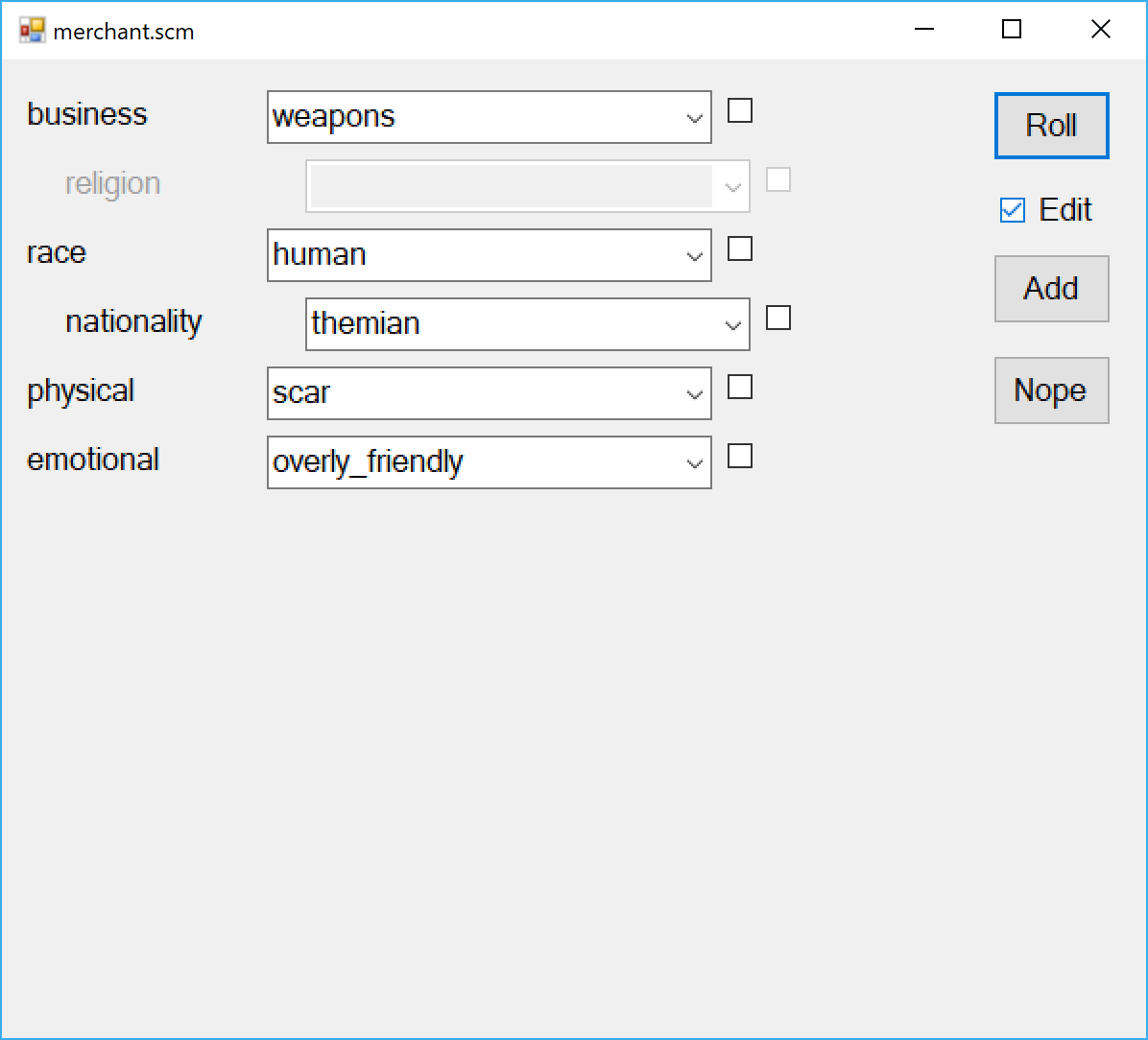
Press the Roll button a few times and you’ll see the values change on screen. So far, so good. But suppose you specifically need a weapons merchant. Just click in the “business” field and type weapons, since that’s one of choices available in business’ domain. Now press Roll:



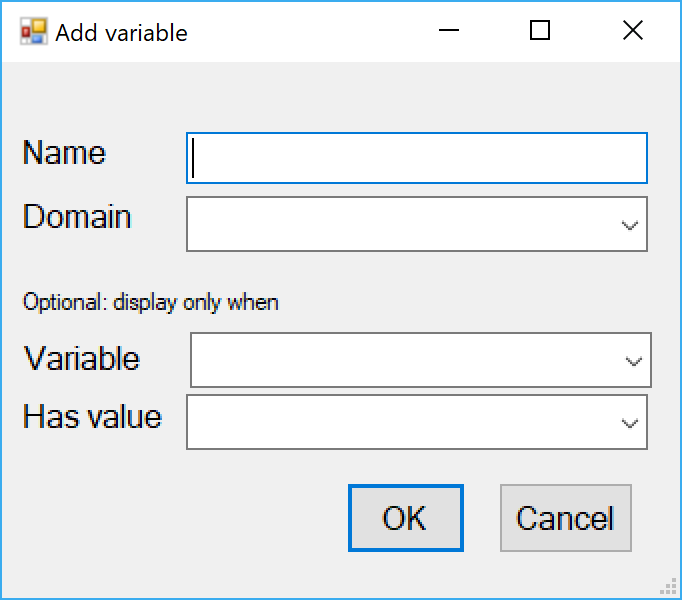
Notice the box next to the business field is checked. That means this variable is **locked** – the generator will work around it and only generate choices consistent with weapons businesses. Just uncheck the box to unlock it and let the generator roam free again.

# Changing the generator

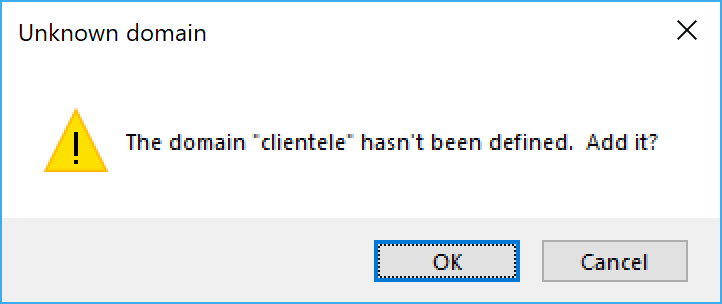
Now check the edit box. The Add and Nope buttons should reappear:



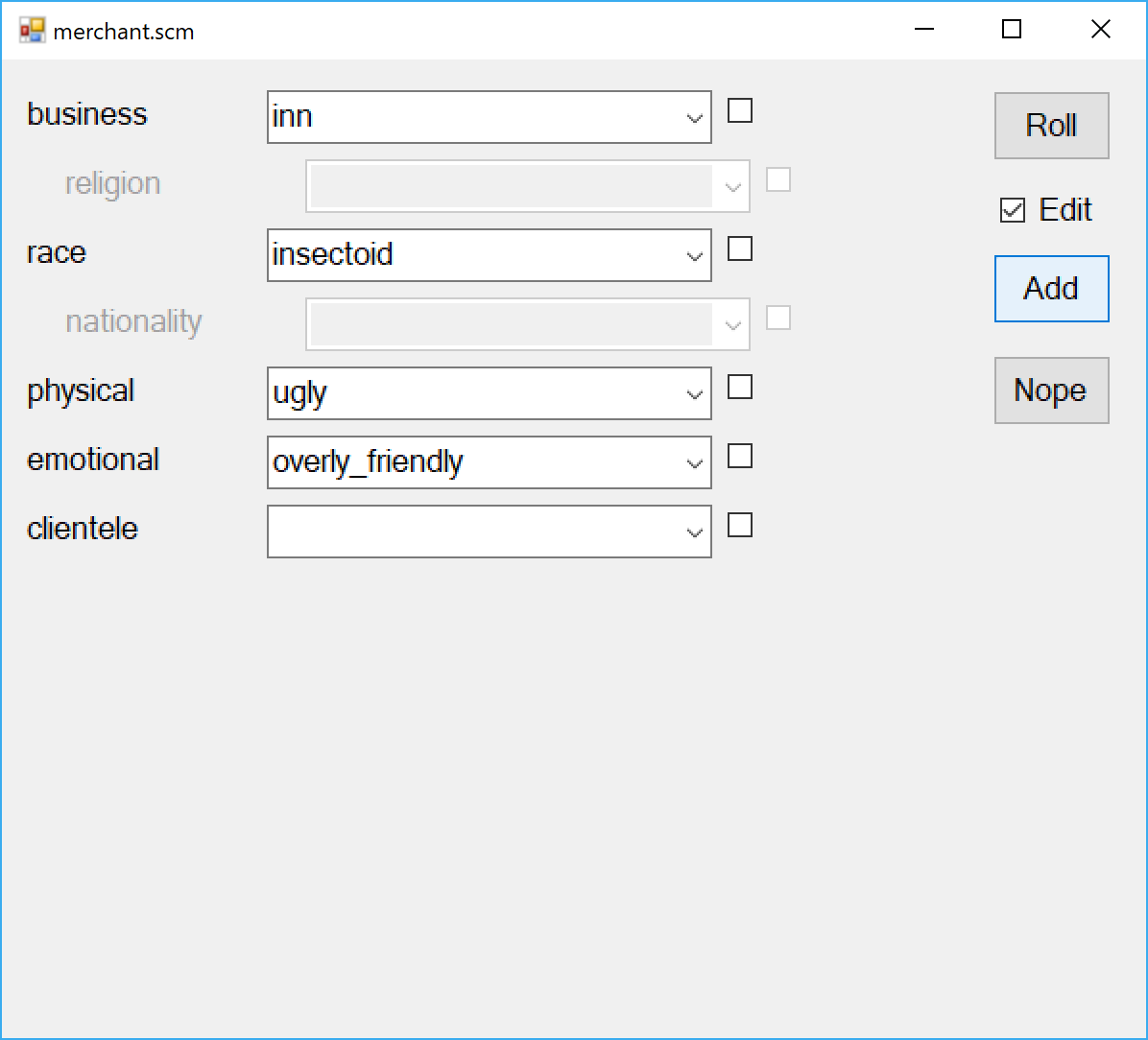
Suppose we want to add a variable to indicate how upscale the merchant’s business is. Click Add, and you’ll see a form to add a new variable:



Enter “clientele” for both name and domain and hit OK. Since there isn’t any domain defined called “clientele”, it will ask to make sure you intend to make one:



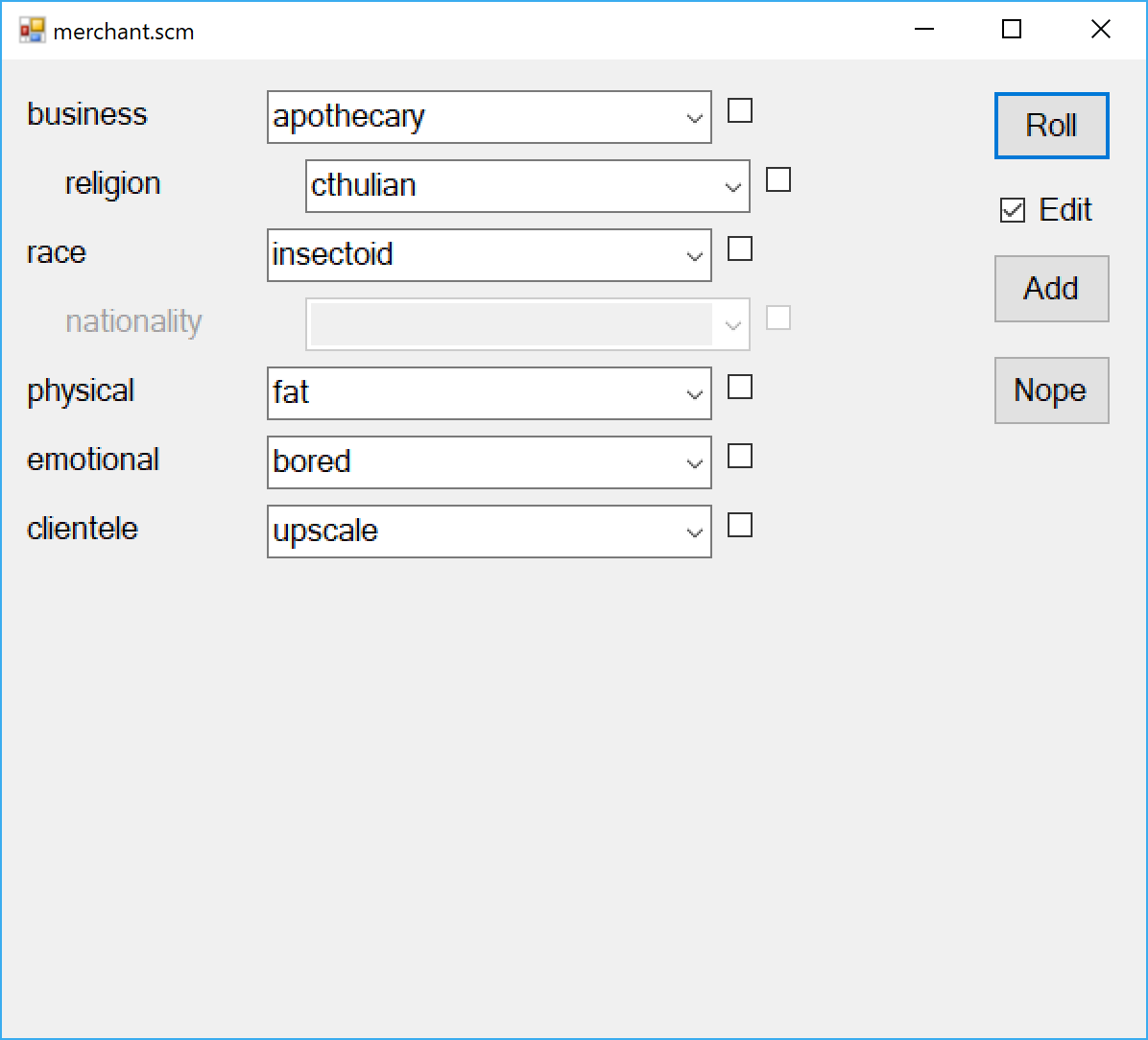
Click okay and you should see something like this:



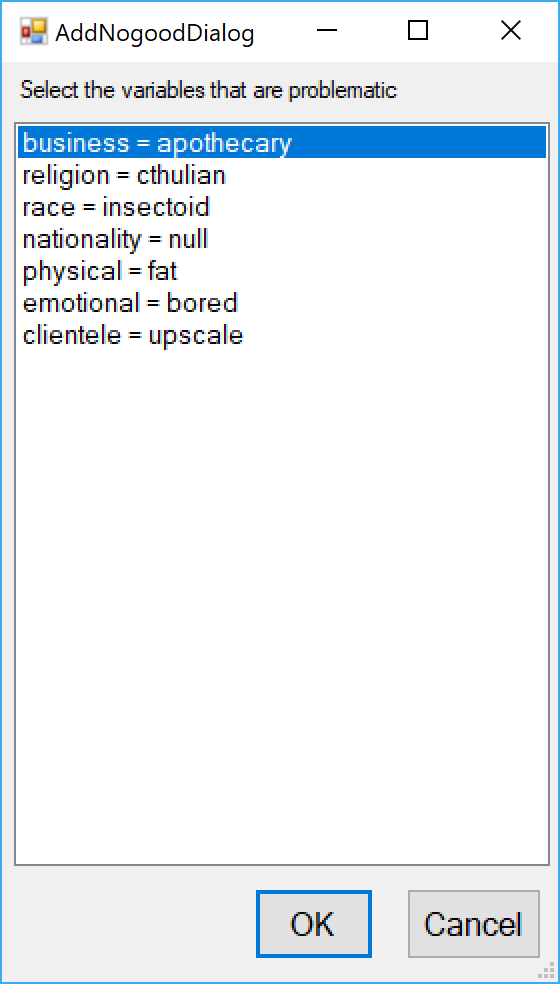
Now there’s a clientele field, but it’s blank, because we haven’t given it any possible values. So click in the clientele field, and type “upscale” and hit return. It will ask if you mean to add it to the domain. Click yes. Now when you hit roll, the generator will assign a value to clientele, but it will always be upscale, because that’s the only value available to it. So now type “downscale”, “shady”, and “desperate”, hitting return each time. It will prompt you each time if you want to add it to the domain, so say yes each time.

# Adding constraints

Now suppose we get a roll like this:



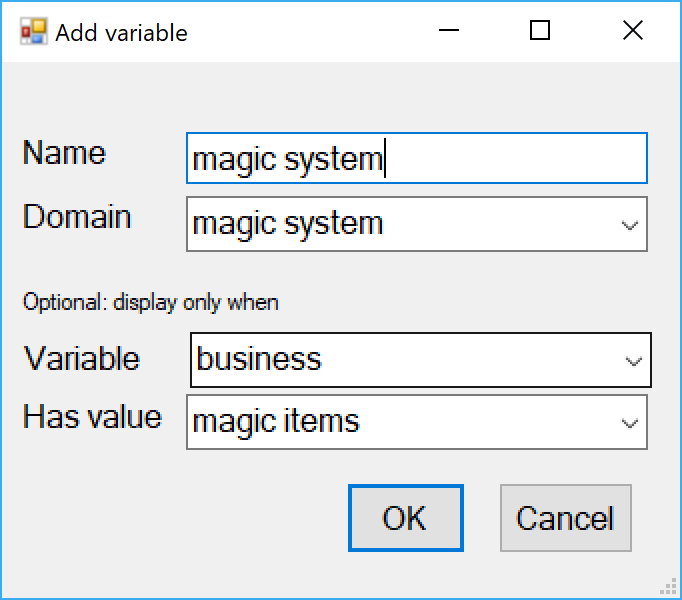
You happen to know that, by definition, there is not such thing as an upscale, Cthulian apothecary and you want it not to generate those in the future. Just press the **Nope** button:



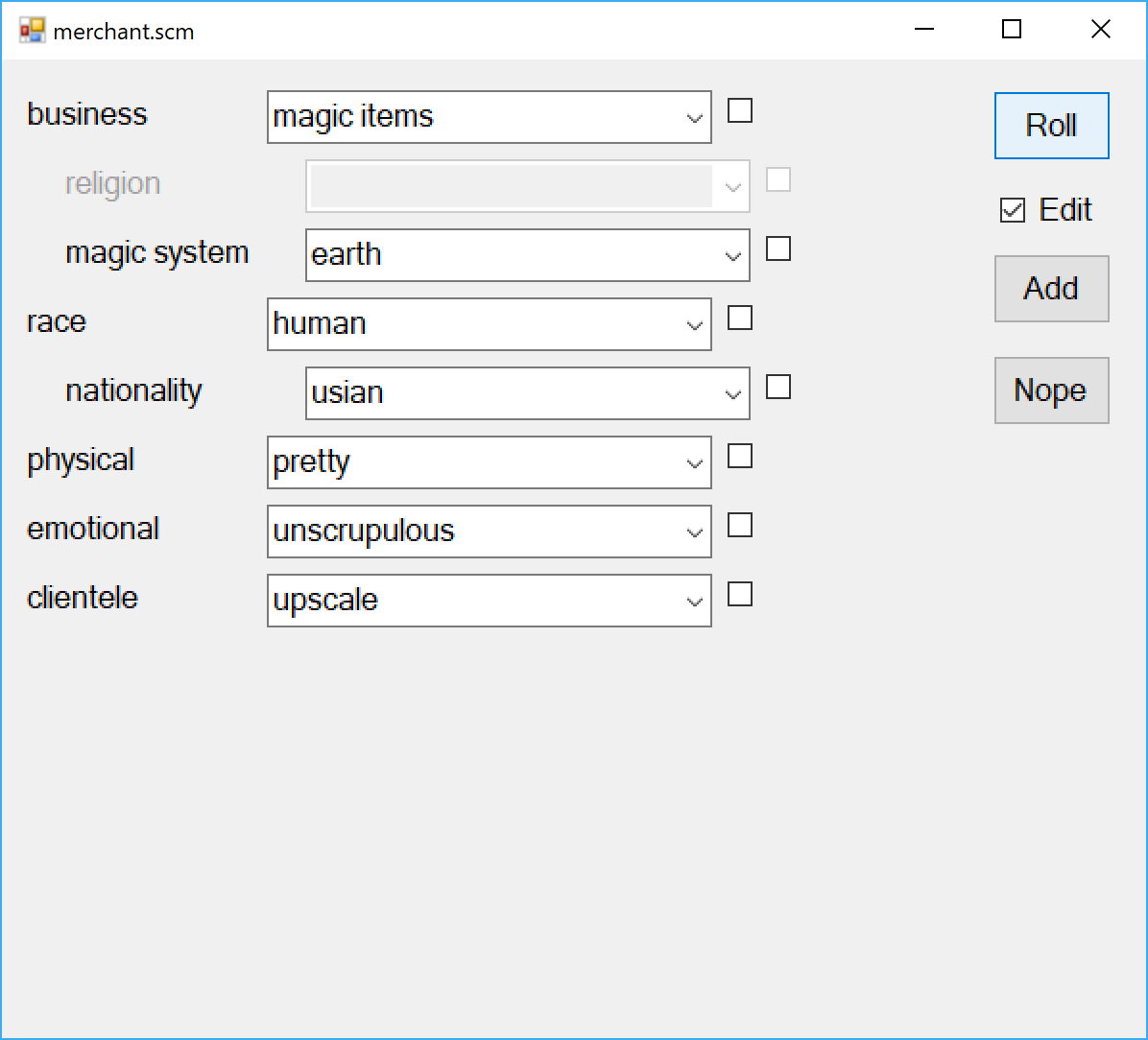
Select the lines that say “business = apothecary”, “religion = cthulian” and “clientele = upscale” and press “OK”. This will teach the generator never to generate any upscale, Cthulian apothecaries again.

# Conditional variables

Now suppose we’re in a world where there are different magic systems (e.g. earth magic vs. fire magic) and a given magical item only sells within a given system. Then we want to add a magic system variable, but only have it defined for businesses that sell magic items. Just press the Add button as usual:



And define the new variable as usual, but under “display only when” fill in “business” for the variable and “magic items” for the value. Hit okay. Enter “earth” and “fire” as values for the magic system domain, and you’re ready to go:



# Commands

Here are the main commands:

* File stuff
  + Control-O  
    Open a file. This will replace what’s in the window now with what’s in the file.
  + Control-S  
    Save generator to a file. If not file name has been specified, it will prompt you to specify where to save it.
  + Control-Shift-S  
    Save generator to an alternate file name.
* Window management
  + Control-N  
    Open a new window, so you can have several generators at once
  + Close box  
    Close the window; closing the last window closes the application.
* Using a generator
  + Roll button  
    Regenerate the values on screen. If some values are locked, they will stay the same and the system will choose
  + Edit checkbox  
    Enables/disables editing of the generator.
  + Typing a new value for a variable  
    Sets the value of the variable and locks it.
  + Checking/unchecking lock boxes  
    Locks/unlocks the variable in question. A locked variable won’t be changed by the system.
  + Control-U  
    Unlocks all variables
* Editing a generator
  + Checking the edit checkbox  
    Enables editing
  + Add button  
    Adds a new variable
  + Typing an unknown value for a variable  
    Lets you add it to the variable’s domain.
  + Nope button

Adds a new nogood combination. The generator will stop generating entities that include the specified combination of variable values

# Editing generator files directly

The graphical user interface for PCGToy is fairly simple. You can’t, for example, remove variables or values from domains, or correct misspellings with it. Fortunately, the generator files are just text files. They are all in S-expression format, which is essentially the less verbose uncle of XML and JSON. The entries are in this form:

* (**domain** *name values* …)  
  Defines a new domain with the specified *name* and set of *values*, all separated by spaces and enclosed in parentheses. If a name or value has a space in it, it should be enclosed in double-quotes.
* (**variable** *name domainname*)  
  Defines a variable with the specified name and domain.
* (**variable** *name domainname* (*conditionvar conditionvalue*))  
  Same, but says that the variable is only defined when *conditionvar* has the value *conditionvalue*.
* (**nogood** (*variable value*) …)  
  Outlaws generating anything in which the specified *variable*s all have the specified *value*s.

Note: there are a few things you can presently do in the files that you can’t do in the UI. In particular, you can have negative conditions. For example:

(variable "magic system" "magic system" (not (business apothecary)))

Means the magic system variable should appear for all businesses *except* apothecaries. Similarly,

(nogood (clientele upscale) (not (emotional “overly friendly”)))

Says that an upscale clientele is incompatible with merchants being anything other than overly friendly.