Getting TK to Work

I originally installed Ruby with Ubuntu's apt-get, but when I installed tk from the website it didn't work at all. I ended up uninstalling Ruby, installing RVM with these instructions

> https://rvm.io/rvm/install

and installing Ruby and tk like this

> $ rvm install ruby-2.3 --with-tcl --with-tk

Then tk worked but the assignment didn't run. Fortunately I found someone on stackoverflow who'd had my exact issue

> https://stackoverflow.com/questions/43011258/ruby-tks-canvas-and-shapes-are-bugging-out

The solution was to paste this code underneath the require in hw6graphics.rb

```
module TkItemConfigOptkeys
   def itemconfig_hash_kv(id, keys, enc_mode = [], conf = [])
      hash_kv(__conv_item_keyonly_opts(id, keys), enc_mode, conf)
   end
end
```

I'll also submit the modified hw6graphics.rb

Challenge – AutoTetris

What it Does

If you press the "a" key, the current piece will be placed automatically in a favorable spot. I can't guaranty that this is the most optimal spot, it's just the program's best guess.

The best way to test it is just to hold down the "a" key and let the game play itself. It takes a while to die this way (definitely way longer than you want to spend marking this assignment) so just take my word that it usually makes it past ten thousand points, but it does eventually die.

How it Works

First, it lists all the possible spots in which the current shape can end up. It only selects a spot if there's actually a path for the block to move into position. An earlier version had it teleporting blocks into closed-off spots.

Then, it uses several heuristics to assign a score to each spot, and moves the block to the best spot. If there are several spots with the same score it picks randomly between them so as not to favor one side of the board. Each heuristic is calculated and weighted individually.

The challenge code is commented. Refer to it for information about the individual heuristics.

Limitations

- When selecting spots, it incorrectly assumes that a shape can only end up at one height per column. This would be easy to fix but I can't be bothered.
- It's still possible to fool it into teleporting a block into an impossible space, but you have to go out of your way to do it.
- It leaves empty squares everywhere, because the heuristics aren't perfect. It is definitely feasible to make the AI play perfectly and never die, but it would be significantly more complex and take me a while.
- I could even keep adding heuristics and changing the weights and probably get closer to correctness, but I had to stop at some point.
- Lots of calculations are repeated and could be optimized away
- This assignment took me way longer than most, so while code quality isn't *bad*, it's not great either.

All in all, I'm pretty impressed at how well it works considering the slapdash nature of the heuristics.