Kameron Mace

8/29/2024

Reading CC1 – 2

Important Principles of Clean Code Naming:

- Intention revealing names
    - The name should answer some questions like: why does it exist? what does it do? how is it used?
    - Should not need to comment the answers to these questions
- Don't be misleading
    - abbreviations can mean different things, for example: HP could mean health points, hypotenuse, hippo, etc... This can be confusing.
- Be wary of similar names
    - words should be meaningfully distinct: don't purposely misspell a word that is already being used, so that it can be used again.
- Pronounceable names
    - Easier to discuss real words
- Classes and Methods
    - Classes should be nouns, methods should be verbs
- Searchable names
    - You can't search "k," there will most likely be different usages of "k" in spellings.
- One word per concept
    - Using multiple different words for one task can lead to confusion, for example: get, fetch, and retrieve
- Context
    - Words can have different meanings: If you take "state" out of context it could mean speaking, or a location, it is important to be clear.

Code Snippets from CS 120 Final Project:

…//

```
  if room == "HallEast":
    if direction.upper() == "N":
      q = requestString("There are two rooms, would you like to go to the
bathroom to the west or the kitchen to the east?(Type ""W"" for bathroom
or ""E"" for kitchen.)")
      if q.lower() == "w":
```

Commented [KM1]: Multiple instances of single letter naming, “q” is not clear to the user.

```
        return "Bathroom"
      if q.lower() == "e":
        return "Kitchen"
```

//...

```
if creature == 0:
    displayCreature()
    decision = requestString("There is a nice-looking creature in the
corner of the room. It offers you a rock. Do you accept the kind gift?
y/n")
    handleCreatureDecision(decision)
    repaint(FloorPlan)
  clearCreature()
  return info
```

//...

```
def copy(source,target,targX,targY):
  targetX=targX
  for sourceX in range (0,getWidth(source)):
    targetY=targY
    for sourceY in range (0,getHeight(source)):
      px=getPixel(source,sourceX,sourceY)
      tx=getPixel(target,targetX,targetY)
      setColor(tx,getColor(px))
      targetY=targetY+1
    targetX=targetX+1
  return (target)
```

Commented [KM2]: creatureDecision would do just fine, handle is unnecessary.

Commented [KM3]: There are four different uses of targ-et , addinng identifiers could be useful here.