Programming Assignment #6: Autograder Visible Test Cases' Inputs

Changelog

- v.1: Initial version.

Part #1

Cases #4 and #6 are hidden.

Case #1

File:

```
79
47
64
20
80
79
```

Case #2

File:

```
-5
18
-14
20
17
18
15
30
35
17
19
-100
20
28
34
27
38
```

Case #3

File:

```
-3
-3
-3
```

Case #5

This test case also reruns case #1, so if you miss case #1, you can't get this one right.

File:

```
90
1000 200
45
87 22
90
```

Part #2

Case #5 is hidden.

Case #1

- Input file:

```
abcdefgh
xyzxyz
wxyza
xyyz
```

- Other parameters: "xyz", `True`
- Expected return value: 2
- Expected output file:

```
xyzxyz
wxyza
```

Case #2

- Input file:

```
abcdefgh
xyzxyz
x y z
wxyza
xyyz
```

- Other parameters: "xyz", `False`
- Expected return value: 3
- Expected output file:

```
abcdefgh
x y z
xyyz
```

Case #3

- Input file:

```
abc
abcd
abcdde
abcddde
```

- Other parameters: "abcde", `False`
- Expected return value: 4
- Expected output file:

```
abc
abcd
abcdde
abcddde
```

Case #4

- Input file:

```
abc
abcd
abcdde
abcddde
```

- Other parameters: "abcde", `True`
- Expected return value: 0
- Expected output file: (empty)

Part #3

Cases #5 and #6 are hidden.

As stated in the directions, "If the function is supposed to return False, then the 2D list will not be checked; in other words, when the autograder expects the return value to be False, it will not check the board at all, and you do not need to undo any changes made to the board."

Case #1

```
e = Entity('T', 2, 1, 2, 4)
board = [['_'] * 4 for i in range(5)]
retval = draw_entity(e, board)
```

Case #2

```
e = Entity('*', 2, 0, 3, 2)
board = [['_'] * 6 for i in range(3)]
retval = draw_entity(e, board)
```

Case #3

```
e1 = Entity('Y', 2, 0, 2, 4)
e2 = Entity('Z', 1, 3, 4, 3)
board = [['_'] * 6 for i in range(7)]
retval = draw_entity(e1, board)
retval = draw_entity(e2, board)
```

Case #4

```
e = Entity('Y', 2, 1, 2, 5)
board = [['_'] * 10 for i in range(5)]
retval = draw_entity(e, board)
```

Part #4.1

Case #4 is hidden.

Case #1

```
b = Building(1, 0)
board = [[' '] * 9 for i in range(10)]
b.draw_on_board(board)
```

Case #2

```
b1 = Building(1, 0)
b2 = Building(2, 6)
board = [[' '] * 9 for i in range(10)]
b1.draw_on_board(board)
b2.draw_on_board(board)
```

Case #3

```
b = Building(0, 0)
board = [[' '] * 6 for i in range(4)]
b.draw_on_board(board)
```

Part #4.2

Case #4 is hidden.

Case #1

```
b = Building(1, 0)
retval = b.contains(2, 0)
# check retval...
retval = b.contains(8, 4)
# check retval...
```

Case #2

```
b = Building(1, 0)
retval = b.contains(10, 20)
# check retval...
retval = b.contains(2, 1)
# check retval...
```

Case #3

```
b = Building(3, 4)
retval = b.contains(3, 5)
# check retval...
retval = b.contains(3, 3)
# check retval...
```

Part #4.3

No cases are hidden.

Each of these cases passes a certain file to the Game initializer and then checks certain attributes (of the instance `g` of `class Game`) such as among the following:

- `type(g.hero)`
- `g.num_objects`
- `g.objects` (size and type of each element)
- `g.buildings` (size and type of each element)
- `g.board`

Case #1a through #1d

- File:

```
20 16
X 3 8
w
a
s
d
o $ 8 14
b 5 2
o $ 17 10
```

Case #2a through #2c

- File:

```
16 16
X 4 9
w
a
s
d
o # 1 1
o # 2 1
o # 3 1
b 4 1
o * 10 1
b 2 10
b 9 11
```

Case #3

- File:

```
10 10
X 8 8
w
a
s
d
o $ 1 1
```

Case #4

- File:

```
14 14
X 2 3
u
l
d
r
o E 5 7
o E 5 8
o E 2 1
b 1 9
o E 12 11
b 4 3
```

Case #5

- File:

```
24 18
X 14 6
w
a
s
d
b 2 2
b 9 2
b 16 2
b 2 7
b 9 7
b 16 7
b 2 12
b 9 12
b 16 12
o $ 1 1
o $ 1 4
o $ 3 6
o $ 8 6
o $ 22 15
o $ 8 16
o $ 10 16
```

Part #4.4

Cases #13 and #14 are hidden.

Relevant input files:

- `input1.txt`:

```
20 16
X 3 8
w
a
s
d
o $ 8 14
b 5 2
o $ 17 10
```

- `input2.txt`:

```
16 16
X 4 9
w
a
s
d
o # 1 1
o # 2 1
o # 3 1
b 4 1
o * 10 1
b 2 10
b 9 11
```

- `input4.txt`:

```
14 14
X 2 3
u
l
d
r
o E 5 7
o E 5 8
o E 2 1
b 1 9
o E 12 11
b 4 3
```

- `input5.txt`:

```
24 18
X 14 6
w
a
s
d
b 2 2
b 9 2
b 16 2
b 2 7
b 9 7
b 16 7
b 2 12
b 9 12
b 16 12
o $ 1 1
o $ 1 4
o $ 3 6
o $ 8 6
o $ 22 15
o $ 8 16
o $ 10 16
```

Case #1

- File: `input1.txt`
- Inputs:

```
a
q
```

Case #2

- File: `input1.txt`
- Inputs:

```
w
w
d
s
end
```

Case #3

- File: `input2.txt`
- Inputs:

```
w
s
d
a
a
a
exit
```

Case #4

- File: `input2.txt`
- Inputs:

```
w
w
w
d
a
d
a
end
```

Case #5

- File: `input1.txt`
- Inputs:

```
a
a
a
q
```

Case #6

- File: `input1.txt`
- Inputs:

```
d
d
d
d
d
d
d
d
d
d
d
d
d
d
d
d
w
w
w
w
w
w
w
w
d
w
a
q
```

Case #7

- File: `input2.txt`
- Inputs:

```
s
s
a
a
a
s
d
s
s
s
s
d
w
s
d
d
d
d
d
d
d
w
a
q
```

Case #8

- File: `input5.txt`
- Inputs:

```
a
w
d
d
s
a
d
w
d
d
d
d
d
d
d
d
s
s
s
s
s
a
a
a
w
s
a
w
s
a
q
```

Case #9

- File: `input1.txt`
- Inputs:

```
s
s
d
d
d
d
d
d
d
d
d
d
d
d
d
d
a
q
```

Case #10

- File: `input4.txt`
- Inputs:

```
d
r
d
d
d
r
r
d
r
r
r
r
r
r
r
d
u
l
d
r
d
d
q
```

Case #11

- File: `input5.txt`

- Inputs:

```
a
a
a
a
a
a
w
w
w
w
w
a
a
a
a
a
a
a
s
s
s
s
s
d
d
d
d
d
d
d
s
s
s
s
s
s
s
s
s
s
d
d
d
d
d
d
d
d
d
d
```

```
d
d
d
d
w
```

Case #12

- File: `input1.txt`
- Inputs:

```
d
d
d
d
d
d
d
d
d
d
d
d
d
d
s
s
w
s
d
a
s
s
s
s
s
a
a
a
a
a
a
a
a
a
```