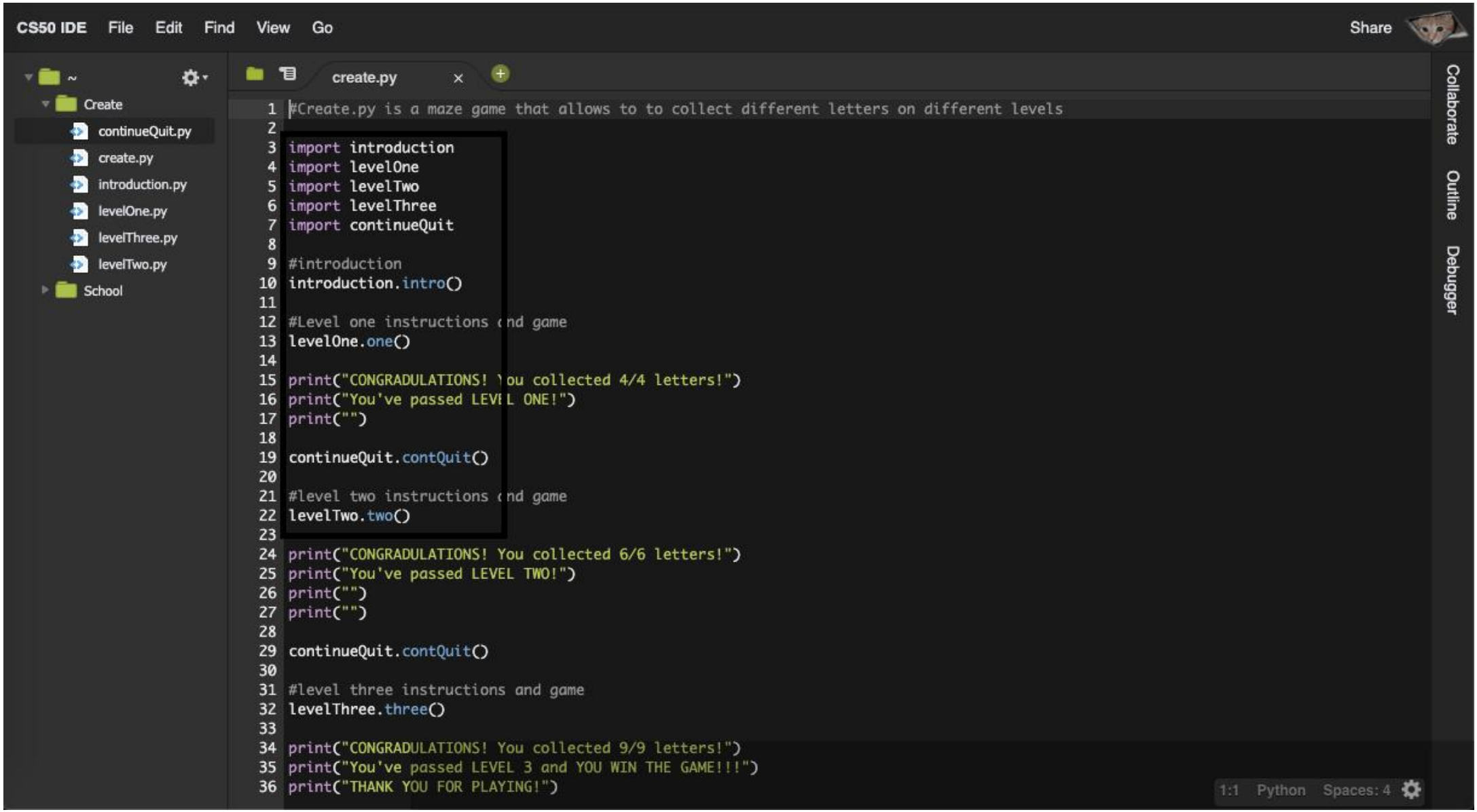
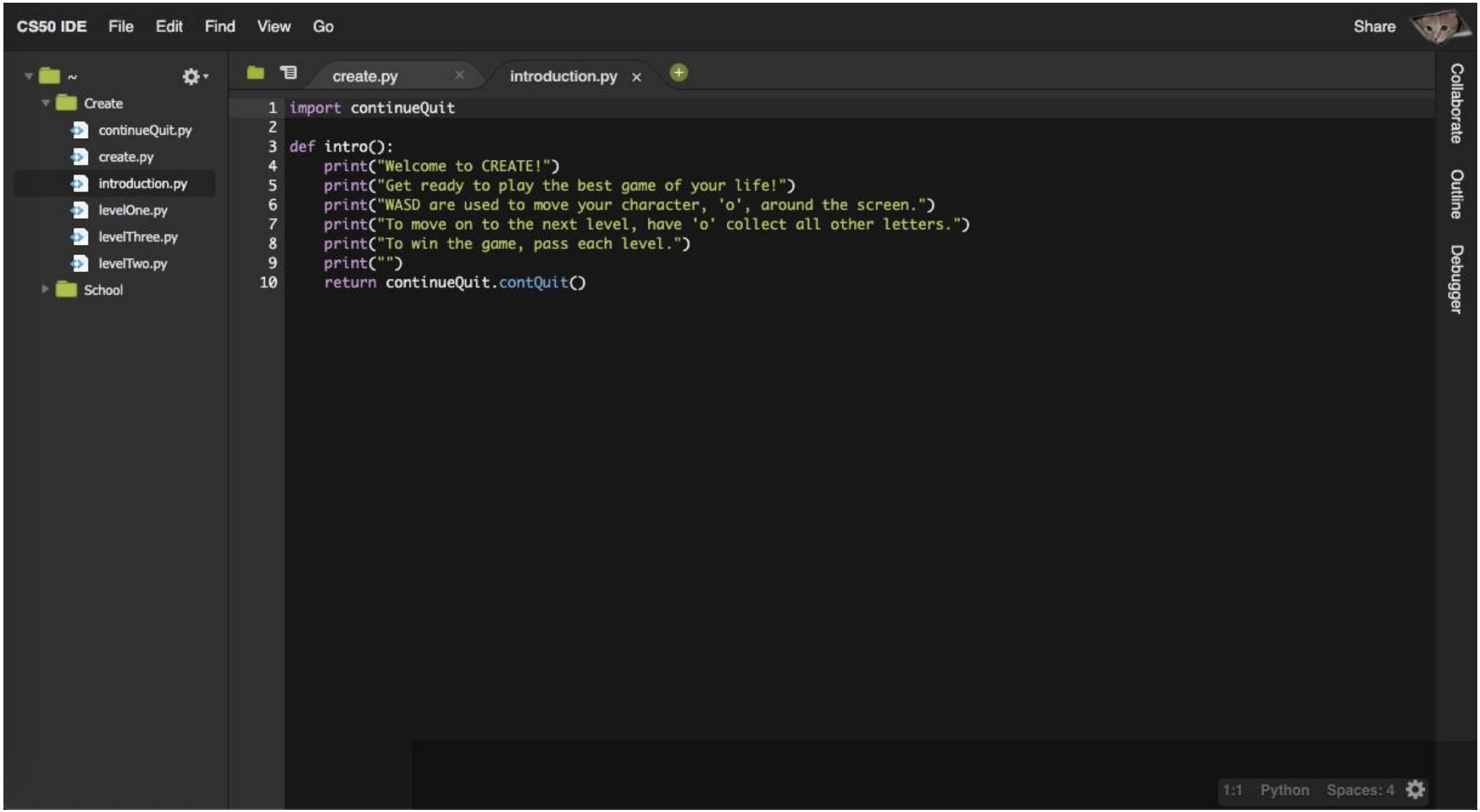
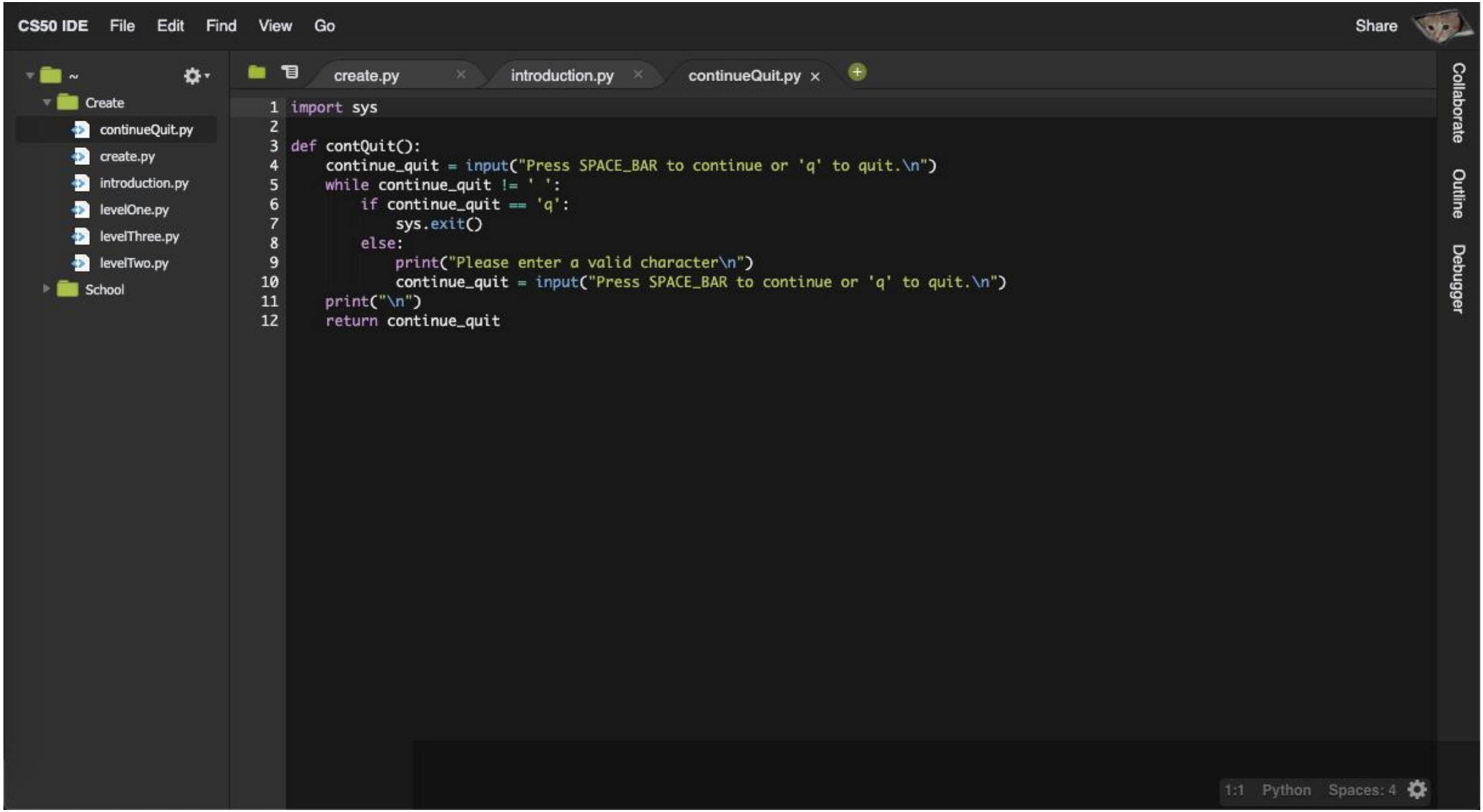
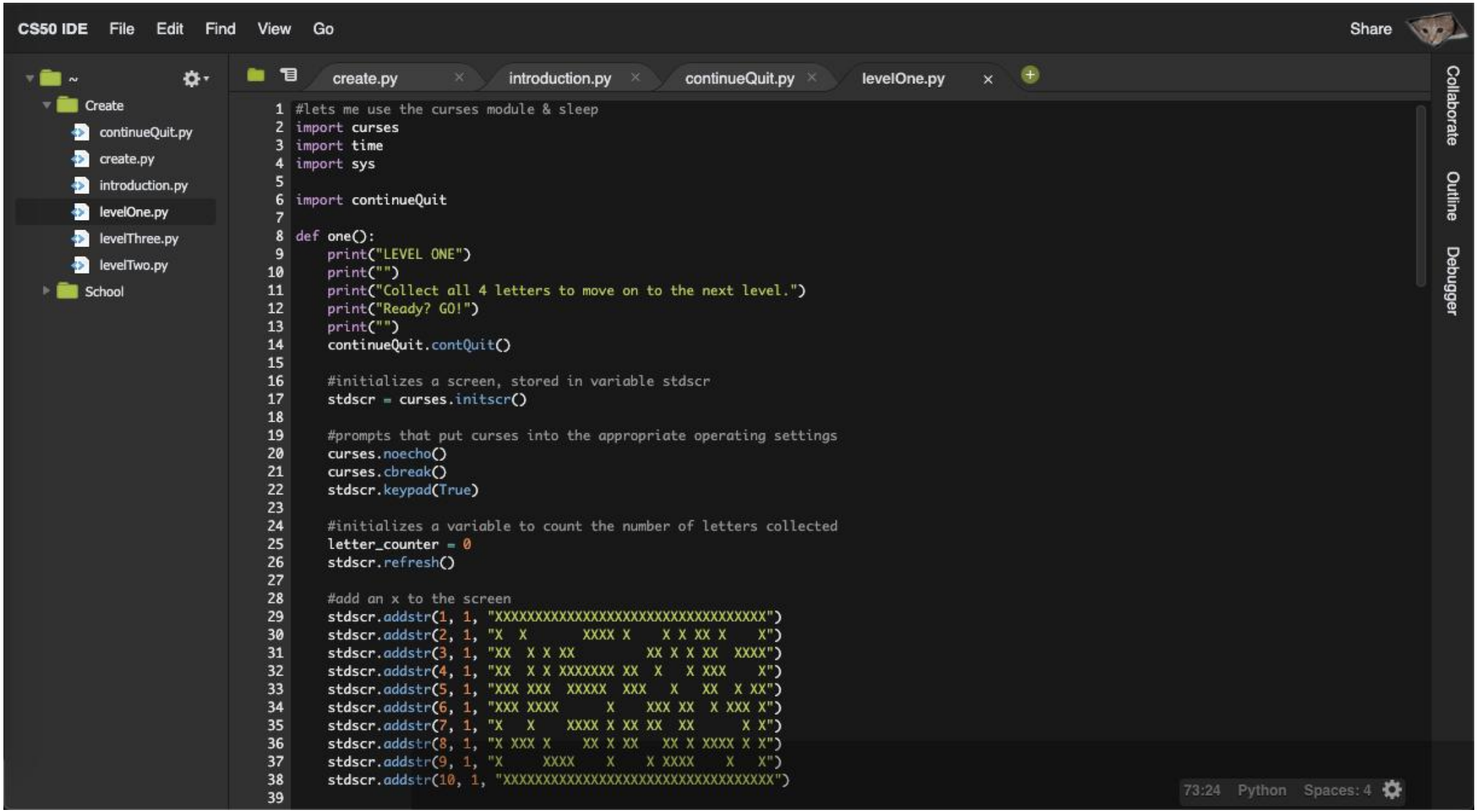
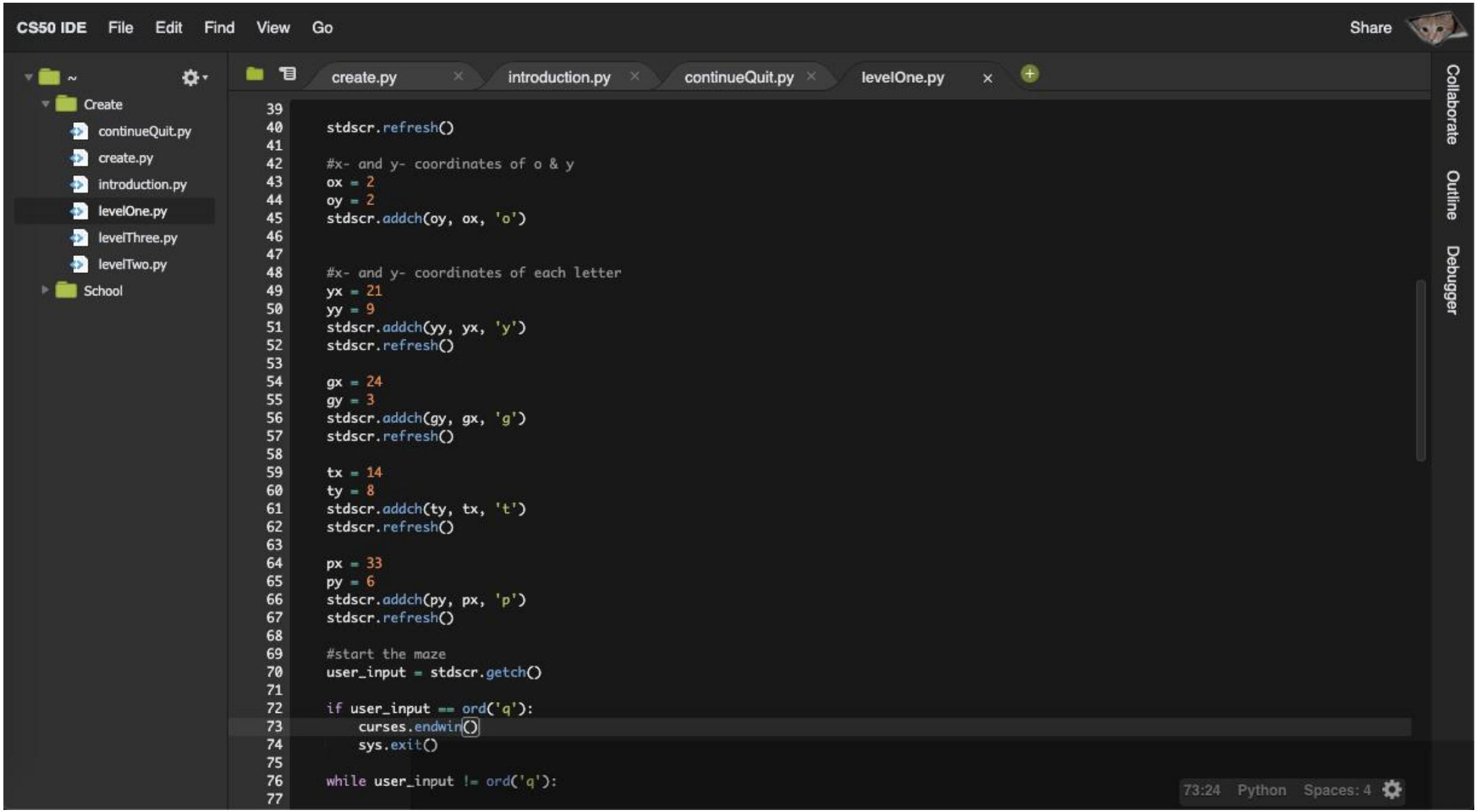
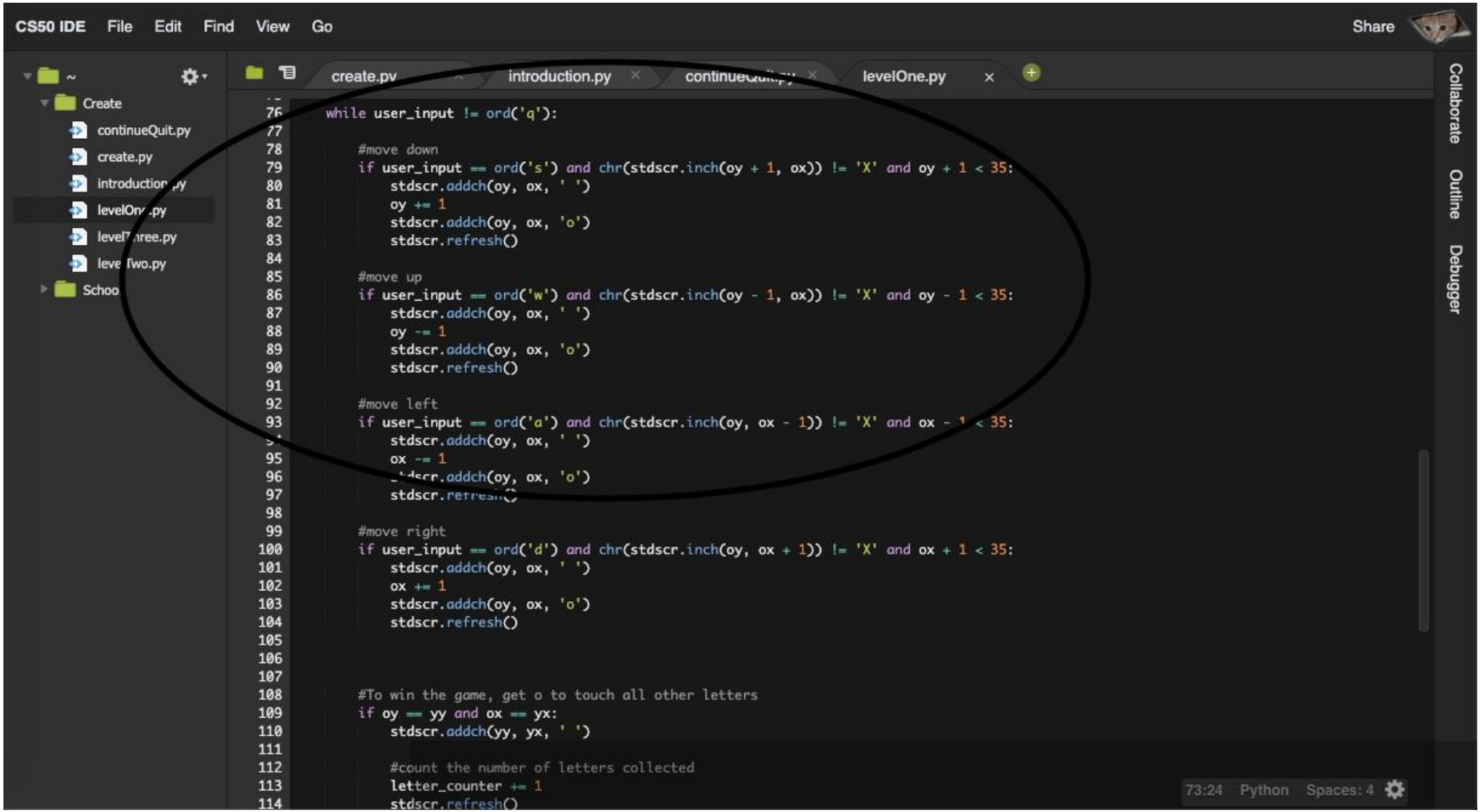

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
Collaborate
Outline
Debugger
#Create.py is a maze game that allows to to collect different letters on different levels
import introduction
import levelOne
import levelTwo
import levelThree
import continueQuit
#introduction
introduction.intro()
#Level one instructions and game
levelOne.one()
print("CONGRADULATIONS! You collected 4/4 letters!")
print("You've passed LEVEL ONE!")
print("")
continueQuit.contQuit()
#level two instructions and game
levelTwo.two()
print("CONGRADULATIONS! You collected 6/6 letters!")
print("You've passed LEVEL TWO!")
print("")
print("")
continueQuit.contQuit()
#level three instructions and game
levelThree.three()
print("CONGRADULATIONS! You collected 9/9 letters!")
print("You've passed LEVEL 3 and YOU WIN THE GAME!!!")
print("THANK YOU FOR PLAYING!")
1:1
Python
Spaces: 4

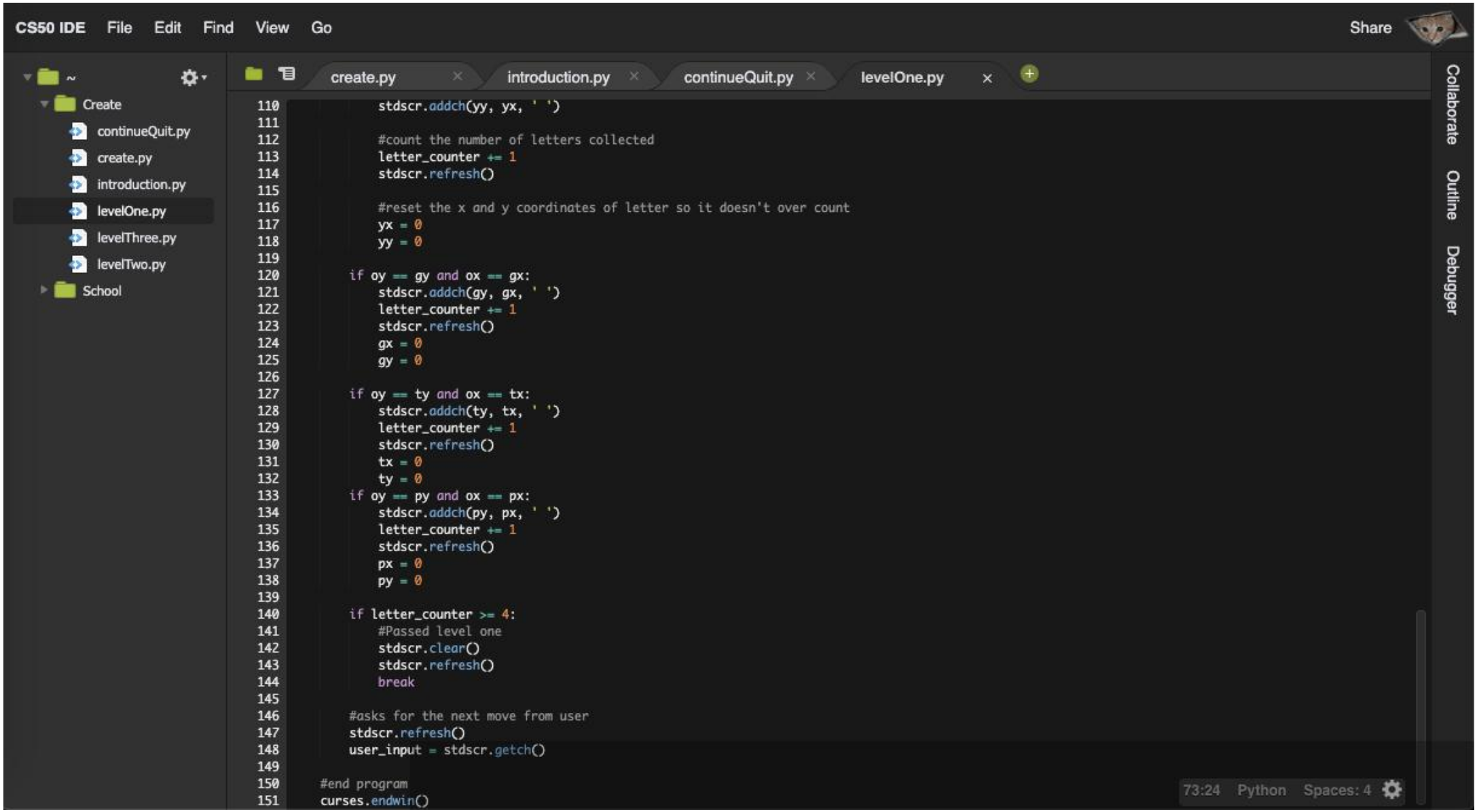

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
import continueQuit
def intro():
print("Welcome to CREATE!")
print("Get ready to play the best game of your life!")
print("WASD are used to move your character, 'o', around the screen.")
print("To move on to the next level, have 'o' collect all other letters.")
print("To win the game, pass each level.")
print("")
return continueQuit.contQuit()
Collaborate
Outline
Debugger
1:1
Python
Spaces: 4

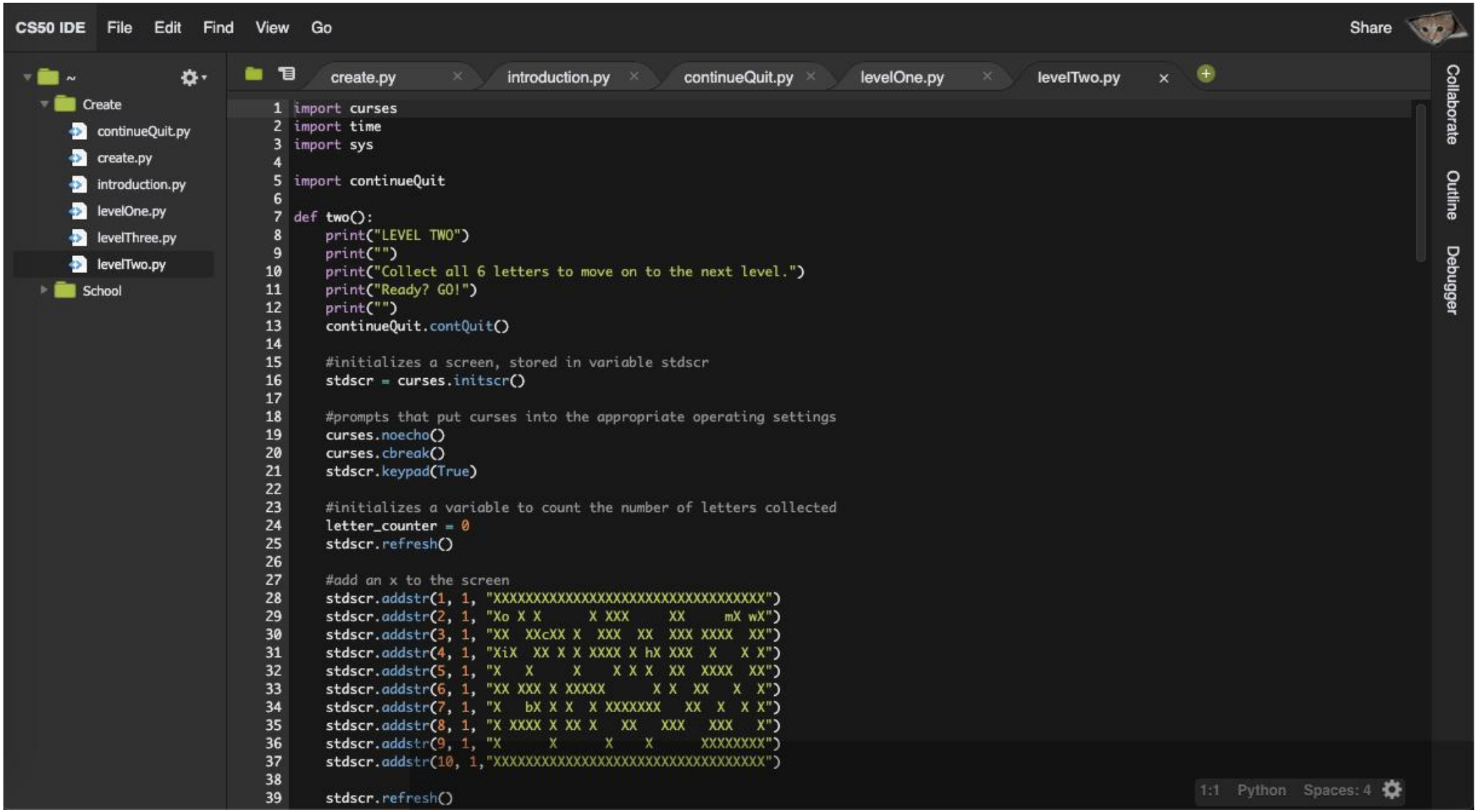

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
import sys
def contQuit():
continue_quit = input("Press SPACE_BAR to continue or 'q' to quit.\n")
while continue_quit != ' ':
if continue_quit == 'q':
sys.exit()
else:
print("Please enter a valid character\n")
continue_quit = input("Press SPACE_BAR to continue or 'q' to quit.\n")
print("\n")
return continue_quit
Collaborate
Outline
Debugger
1:1
Python
Spaces: 4

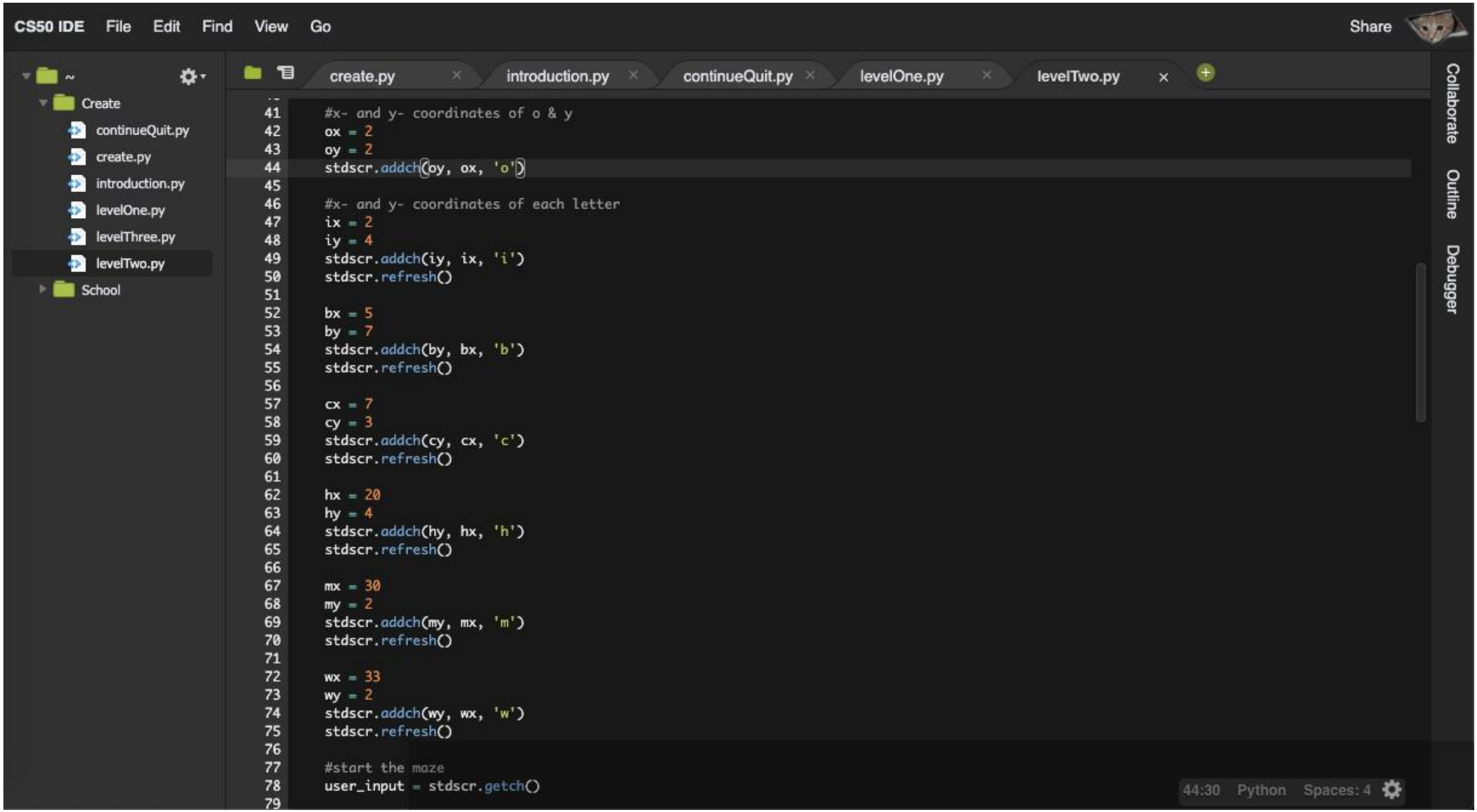

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
Collaborate
Outline
Debugger
#lets me use the curses module & sleep
import curses
import time
import sys
import continueQuit
def one():
print("LEVEL ONE")
print("")
print("Collect all 4 letters to move on to the next level.")
print("Ready? GO!")
print("")
continueQuit.contQuit()
#initializes a screen, stored in variable stdscr
stdscr = curses.initscr()
#prompts that put curses into the appropriate operating settings
curses.noecho()
curses.cbreak()
stdscr.keypad(True)
#initializes a variable to count the number of letters collected
letter_counter = 0
stdscr.refresh()
#add an x to the screen
stdscr.addstr(1, 1, "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
stdscr.addstr(2, 1, "X X XXXX X X X XX X X")
stdscr.addstr(3, 1, "XX X X XX XX X X XX XXXX")
stdscr.addstr(4, 1, "XX X X XXXXXXX XX X X XXX X")
stdscr.addstr(5, 1, "XXX XXX XXXXX XXX X XX X XX")
stdscr.addstr(6, 1, "XXX XXXX X XXX XX X XXX X")
stdscr.addstr(7, 1, "X X XXXX X XX XX XX X X")
stdscr.addstr(8, 1, "X XXX X XX X XX XX X XXXX X X")
stdscr.addstr(9, 1, "X XXXX X X XXXX X X")
stdscr.addstr(10, 1, "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
73:24
Python
Spaces: 4

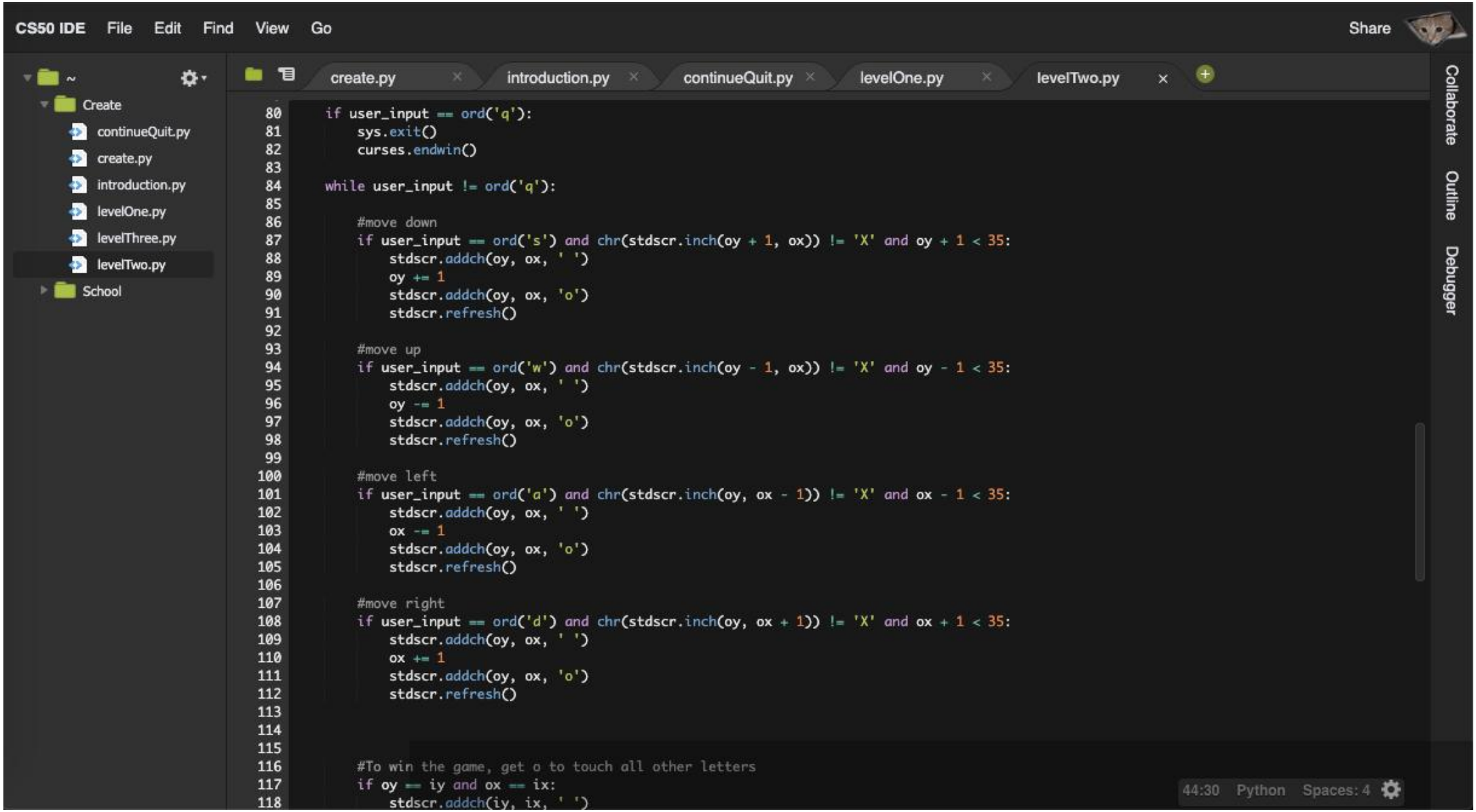

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
Collaborate
Outline
Debugger
39
40 stdscr.refresh()
41
42 #x- and y- coordinates of o & y
43 ox = 2
44 oy = 2
45 stdscr.addch(oy, ox, 'o')
46
47
48 #x- and y- coordinates of each letter
49 yx = 21
50 yy = 9
51 stdscr.addch(yy, yx, 'y')
52 stdscr.refresh()
53
54 gx = 24
55 gy = 3
56 stdscr.addch(gy, gx, 'g')
57 stdscr.refresh()
58
59 tx = 14
60 ty = 8
61 stdscr.addch(ty, tx, 't')
62 stdscr.refresh()
63
64 px = 33
65 py = 6
66 stdscr.addch(py, px, 'p')
67 stdscr.refresh()
68
69 #start the maze
70 user_input = stdscr.getch()
71
72 if user_input == ord('q'):
73 curses.endwin()
74 sys.exit()
75
76 while user_input != ord('q'):
77
73:24
Python
Spaces: 4


CS50 IDE
File
Edit
Find
View
Go
Share
Create
continueQuit.py
create.py
levelTwo.py
create.py
introduction.py
levelOne.py
while user_input != ord('q'):
#move down
if user_input == ord('s') and chr(stdscr.inch(oy + 1, ox)) != 'X' and oy + 1 < 35:
stdscr.addch(oy, ox, ' ')
oy += 1
stdscr.addch(oy, ox, 'o')
stdscr.refresh()
#move up
if user_input == ord('w') and chr(stdscr.inch(oy - 1, ox)) != 'X' and oy - 1 < 35:
stdscr.addch(oy, ox, ' ')
oy -= 1
stdscr.addch(oy, ox, 'o')
stdscr.refresh()
#move left
if user_input == ord('a') and chr(stdscr.inch(oy, ox - 1)) != 'X' and ox - 1 < 35:
stdscr.addch(oy, ox, ' ')
ox -= 1
stdscr.addch(oy, ox, 'o')
#move right
if user_input == ord('d') and chr(stdscr.inch(oy, ox + 1)) != 'X' and ox + 1 < 35:
stdscr.addch(oy, ox, ' ')
ox += 1
stdscr.addch(oy, ox, 'o')
stdscr.refresh()
#To win the game, get o to touch all other letters
if oy == yy and ox == yx:
stdscr.addch(yy, yx, ' ')
#count the number of letters collected
letter_counter += 1
stdscr.refresh()
Collaborate
Outline
Debugger
73:24
Python
Spaces: 4

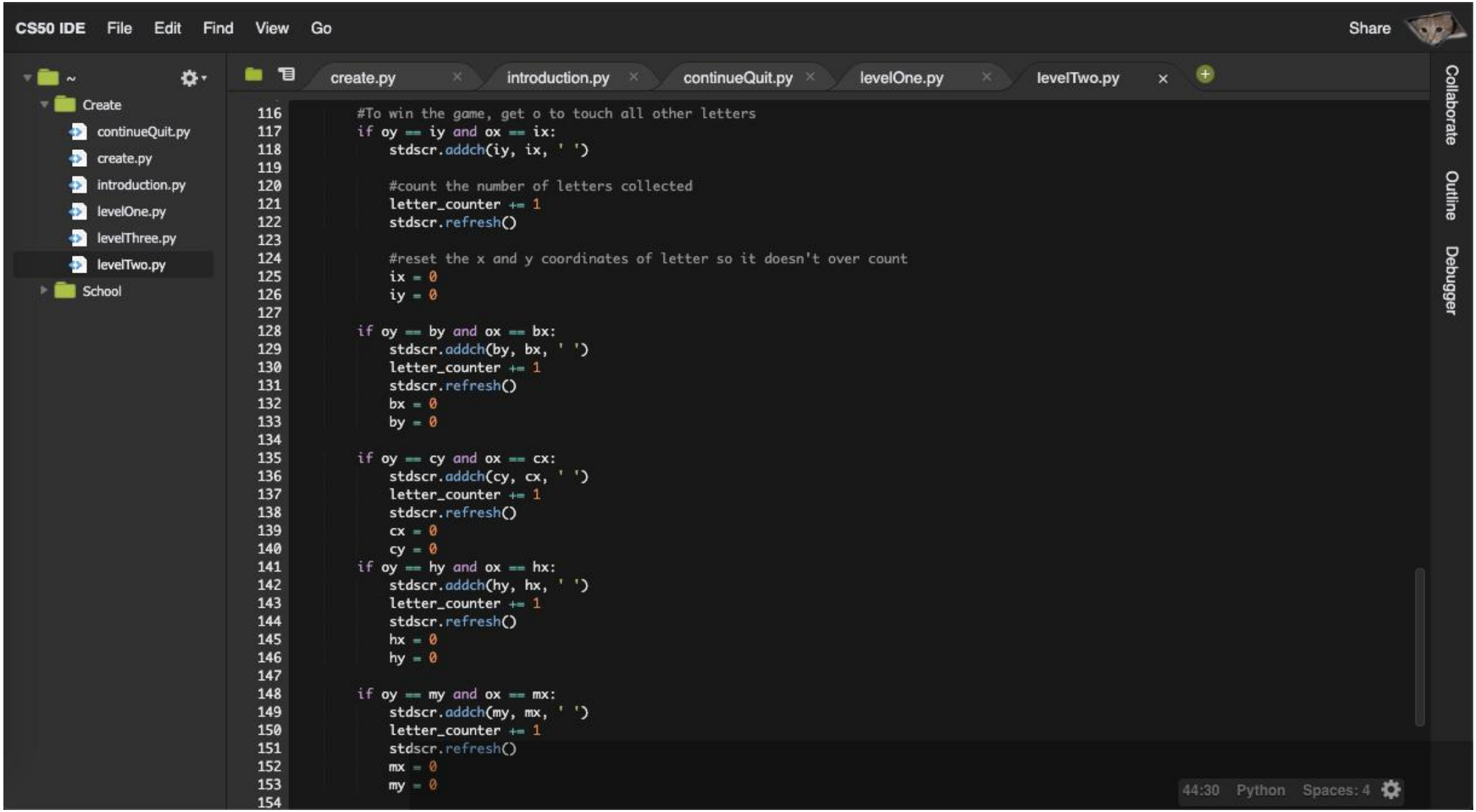

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
Collaborate
Outline
Debugger
stdscr.addch(yy, yx, ' ')
#count the number of letters collected
letter_counter += 1
stdscr.refresh()
#reset the x and y coordinates of letter so it doesn't over count
yx = 0
yy = 0
if oy == gy and ox == gx:
stdscr.addch(gy, gx, ' ')
letter_counter += 1
stdscr.refresh()
gx = 0
gy = 0
if oy == ty and ox == tx:
stdscr.addch(ty, tx, ' ')
letter_counter += 1
stdscr.refresh()
tx = 0
ty = 0
if oy == py and ox == px:
stdscr.addch(py, px, ' ')
letter_counter += 1
stdscr.refresh()
px = 0
py = 0
if letter_counter >= 4:
#Passed level one
stdscr.clear()
stdscr.refresh()
break
#asks for the next move from user
stdscr.refresh()
user_input = stdscr.getch()
#end program
curses.endwin()
73:24
Python
Spaces: 4

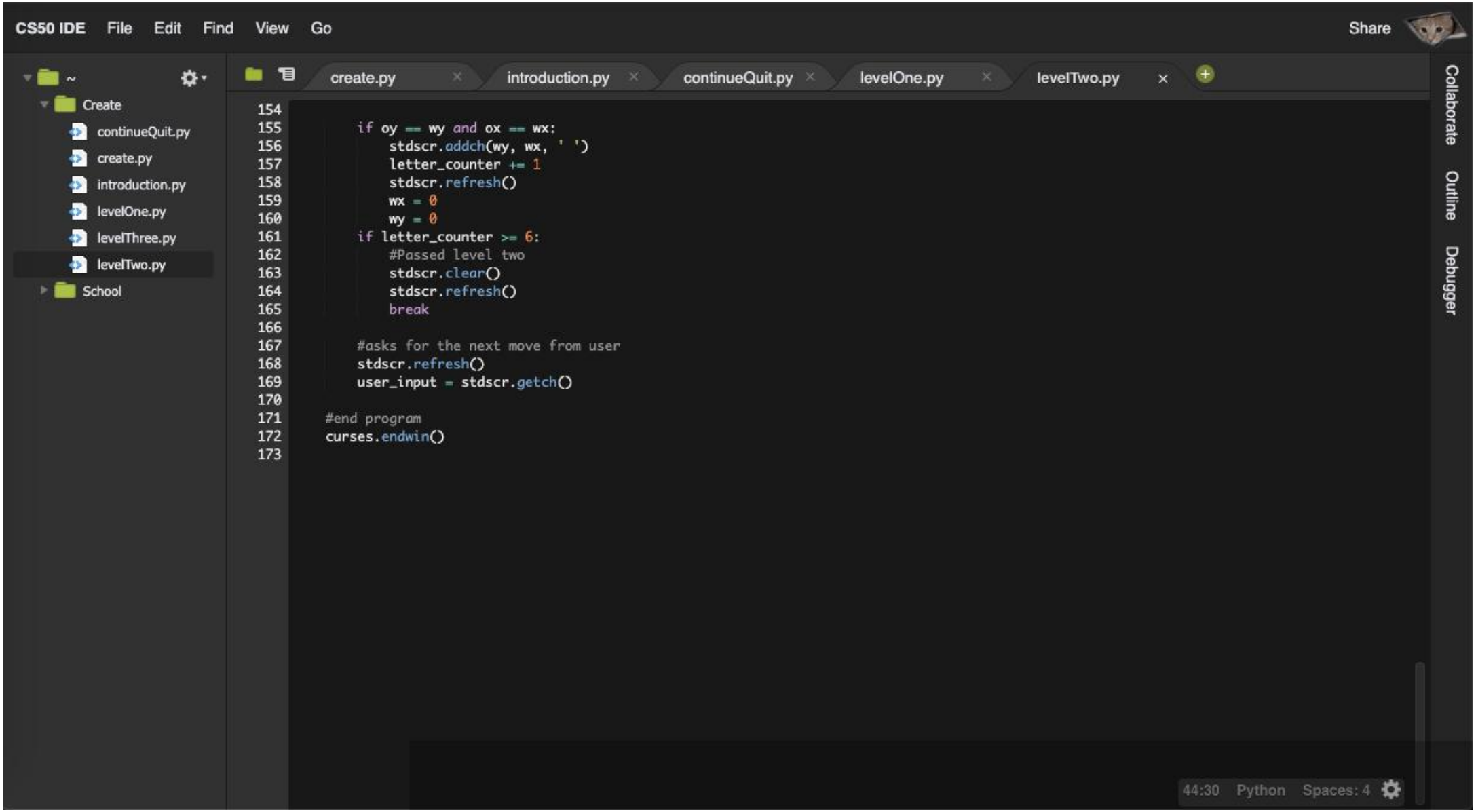

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
Collaborate
Outline
Debugger
1 import curses
2 import time
3 import sys
4
5 import continueQuit
6
7 def two():
8 print("LEVEL TWO")
9 print("")
10 print("Collect all 6 letters to move on to the next level.")
11 print("Ready? GO!")
12 print("")
13 continueQuit.contQuit()
14
15 #initializes a screen, stored in variable stdscr
16 stdscr = curses.initscr()
17
18 #prompts that put curses into the appropriate operating settings
19 curses.noecho()
20 curses.cbreak()
21 stdscr.keypad(True)
22
23 #initializes a variable to count the number of letters collected
24 letter_counter = 0
25 stdscr.refresh()
26
27 #add an x to the screen
28 stdscr.addstr(1, 1, "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
29 stdscr.addstr(2, 1, "Xo X X X XXX XX mX wX")
30 stdscr.addstr(3, 1, "XX XXcXX X XXX XX XXX XXXX XX")
31 stdscr.addstr(4, 1, "XiX XX X X XXXX X hX XXX X X X")
32 stdscr.addstr(5, 1, "X X X X X X XX XXXX XX")
33 stdscr.addstr(6, 1, "XX XXX X XXXXX X X XX X X")
34 stdscr.addstr(7, 1, "X bX X X X XXXXXXX XX X X X")
35 stdscr.addstr(8, 1, "X XXXX X XX X XX XXX XXX X")
36 stdscr.addstr(9, 1, "X X X X XXXXXXXX")
37 stdscr.addstr(10, 1,"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
38
39 stdscr.refresh()
1:1 Python Spaces: 4

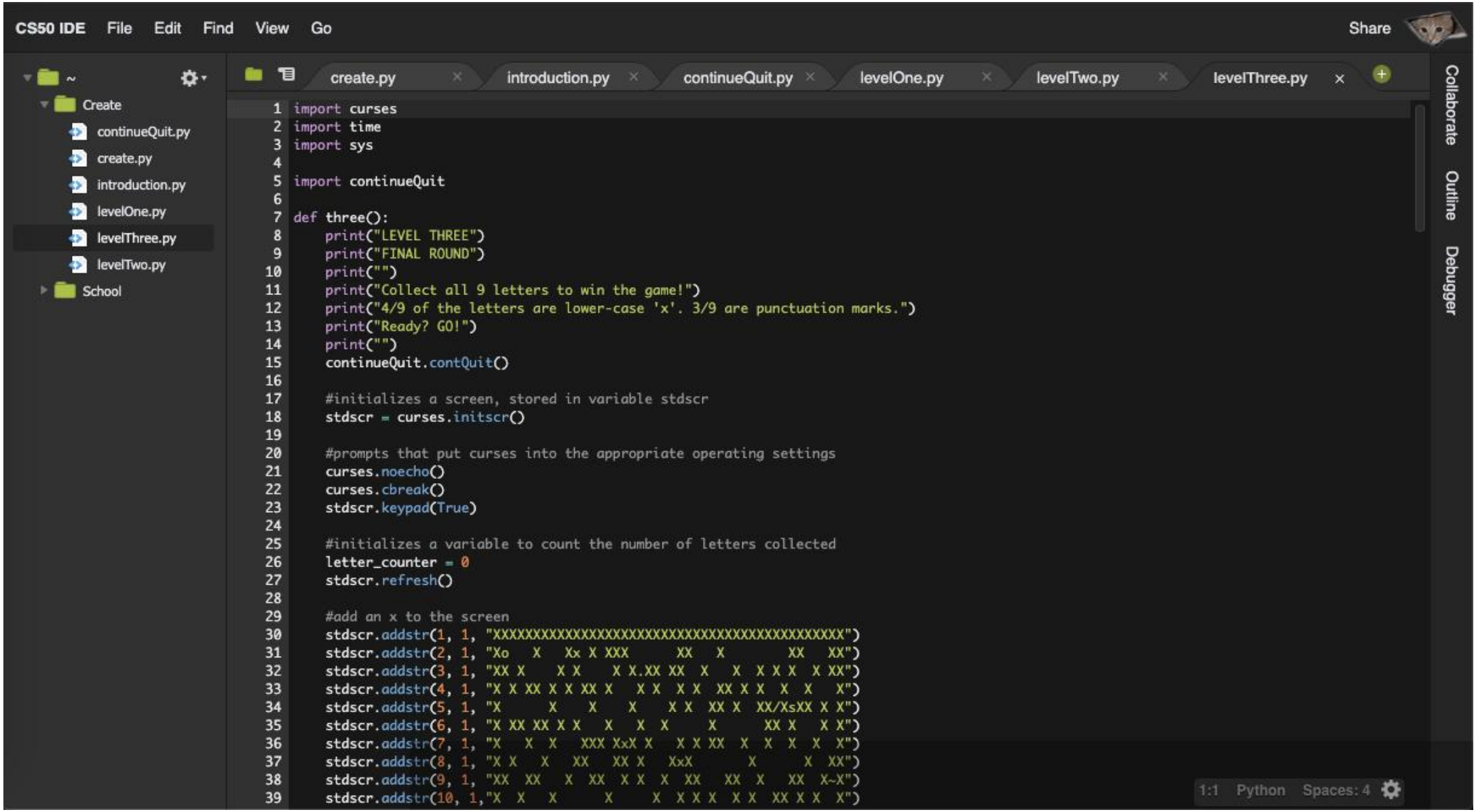

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
#x- and y- coordinates of o & y
ox = 2
oy = 2
stdscr.addch(oy, ox, 'o')
#x- and y- coordinates of each letter
ix = 2
iy = 4
stdscr.addch(iy, ix, 'i')
stdscr.refresh()
bx = 5
by = 7
stdscr.addch(by, bx, 'b')
stdscr.refresh()
cx = 7
cy = 3
stdscr.addch(cy, cx, 'c')
stdscr.refresh()
hx = 20
hy = 4
stdscr.addch(hy, hx, 'h')
stdscr.refresh()
mx = 30
my = 2
stdscr.addch(my, mx, 'm')
stdscr.refresh()
wx = 33
wy = 2
stdscr.addch(wy, wx, 'w')
stdscr.refresh()
#start the maze
user_input = stdscr.getch()
44:30
Python
Spaces: 4
Collaborate
Outline
Debugger

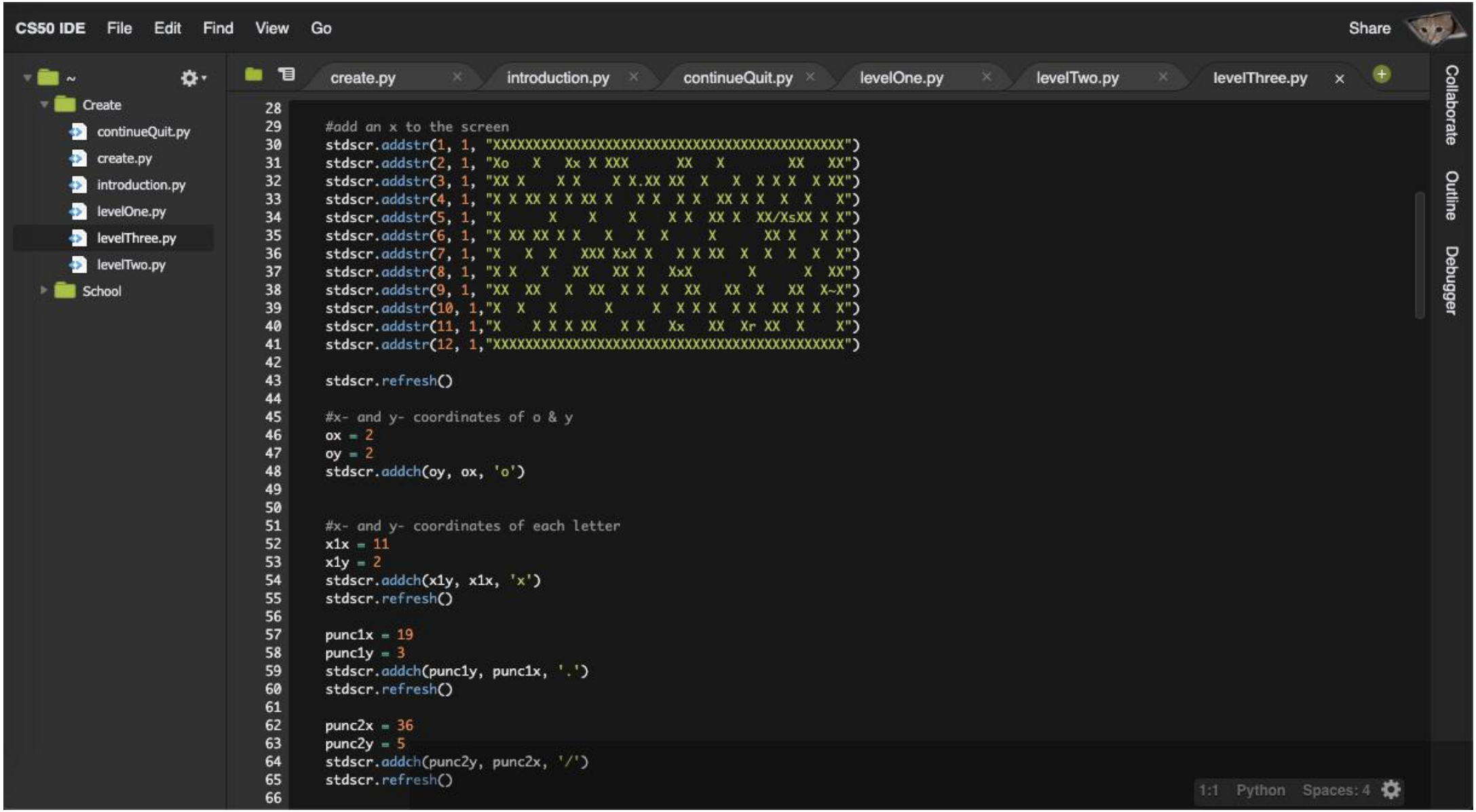


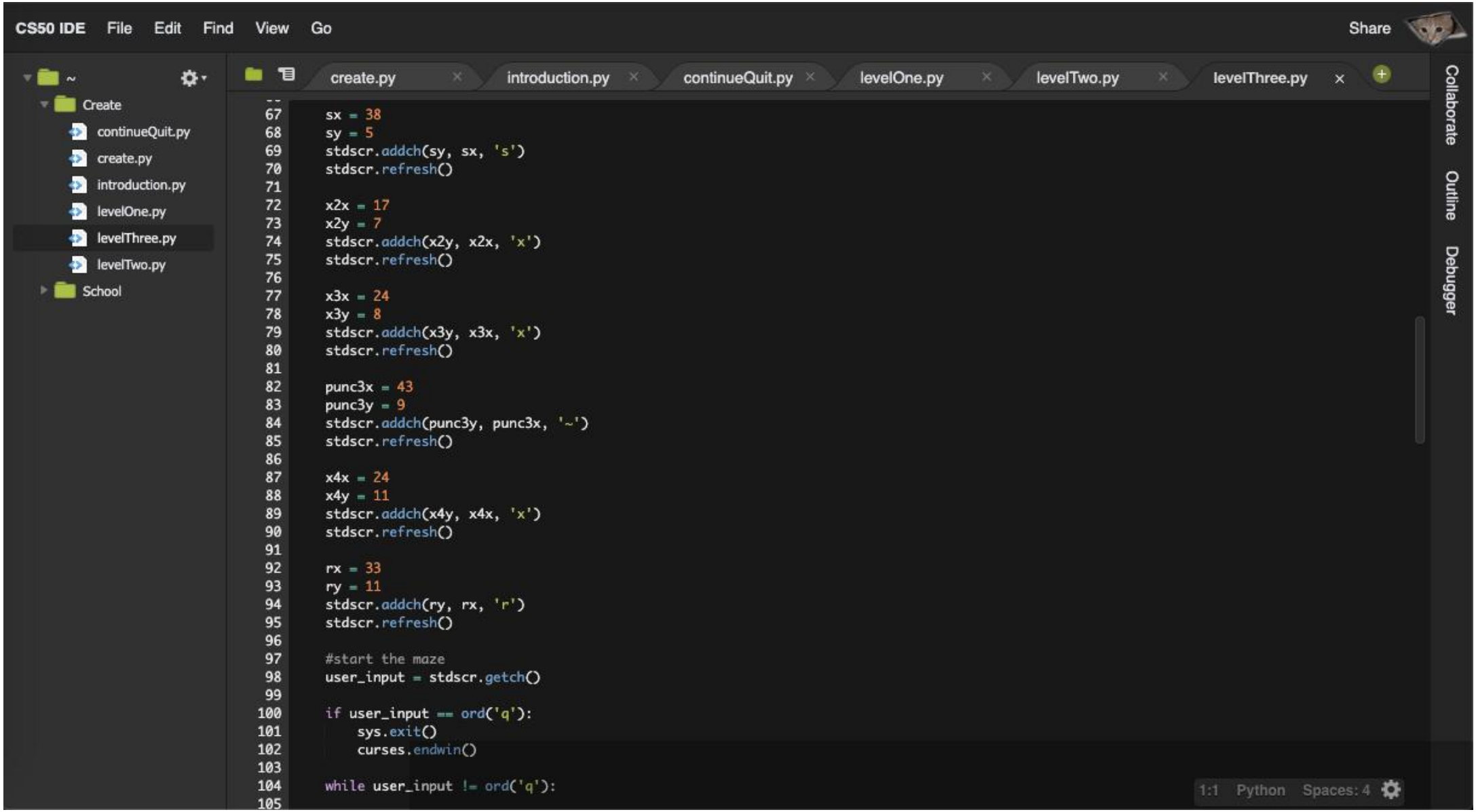

```
80  if user_input == ord('q'):
81      sys.exit()
82      curses.endwin()
83
84  while user_input != ord('q'):
85
86      #move down
87      if user_input == ord('s') and chr(stdscr.inch(oy + 1, ox)) != 'X' and oy + 1 < 35:
88          stdscr.addch(oy, ox, ' ')
89          oy += 1
90          stdscr.addch(oy, ox, 'o')
91          stdscr.refresh()
92
93      #move up
94      if user_input == ord('w') and chr(stdscr.inch(oy - 1, ox)) != 'X' and oy - 1 < 35:
95          stdscr.addch(oy, ox, ' ')
96          oy -= 1
97          stdscr.addch(oy, ox, 'o')
98          stdscr.refresh()
99
100     #move left
101     if user_input == ord('a') and chr(stdscr.inch(oy, ox - 1)) != 'X' and ox - 1 < 35:
102         stdscr.addch(oy, ox, ' ')
103         ox -= 1
104         stdscr.addch(oy, ox, 'o')
105         stdscr.refresh()
106
107     #move right
108     if user_input == ord('d') and chr(stdscr.inch(oy, ox + 1)) != 'X' and ox + 1 < 35:
109         stdscr.addch(oy, ox, ' ')
110         ox += 1
111         stdscr.addch(oy, ox, 'o')
112         stdscr.refresh()
113
114
115
116     #To win the game, get o to touch all other letters
117     if oy == iy and ox == ix:
118         stdscr.addch(iy, ix, ' ')
```

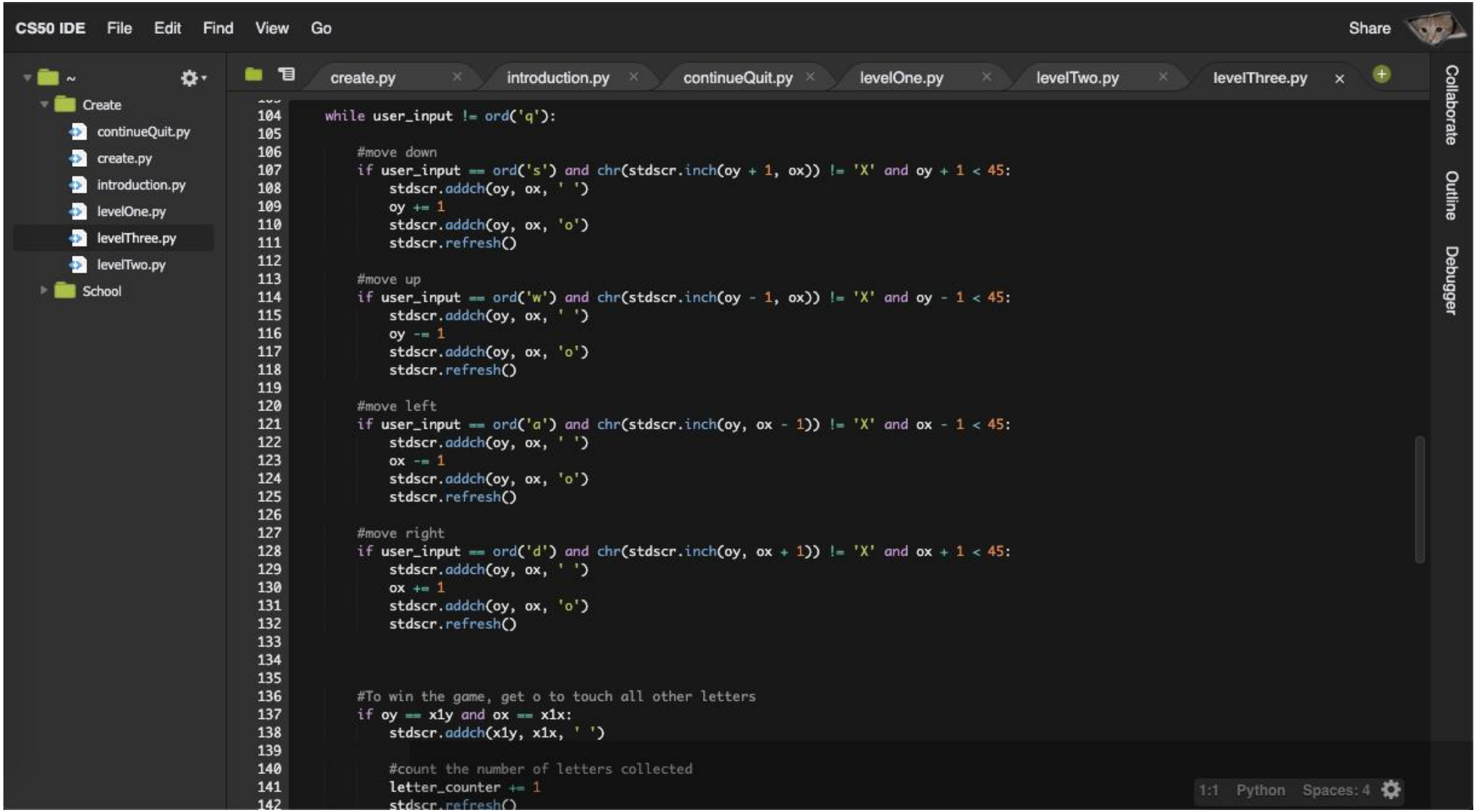

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
Collaborate
Outline
Debugger
#To win the game, get o to touch all other letters
if oy == iy and ox == ix:
stdscr.addch(iy, ix, ' ')
#count the number of letters collected
letter_counter += 1
stdscr.refresh()
#reset the x and y coordinates of letter so it doesn't over count
ix = 0
iy = 0
if oy == by and ox == bx:
stdscr.addch(by, bx, ' ')
letter_counter += 1
stdscr.refresh()
bx = 0
by = 0
if oy == cy and ox == cx:
stdscr.addch(cy, cx, ' ')
letter_counter += 1
stdscr.refresh()
cx = 0
cy = 0
if oy == hy and ox == hx:
stdscr.addch(hy, hx, ' ')
letter_counter += 1
stdscr.refresh()
hx = 0
hy = 0
if oy == my and ox == mx:
stdscr.addch(my, mx, ' ')
letter_counter += 1
stdscr.refresh()
mx = 0
my = 0
44:30
Python
Spaces: 4

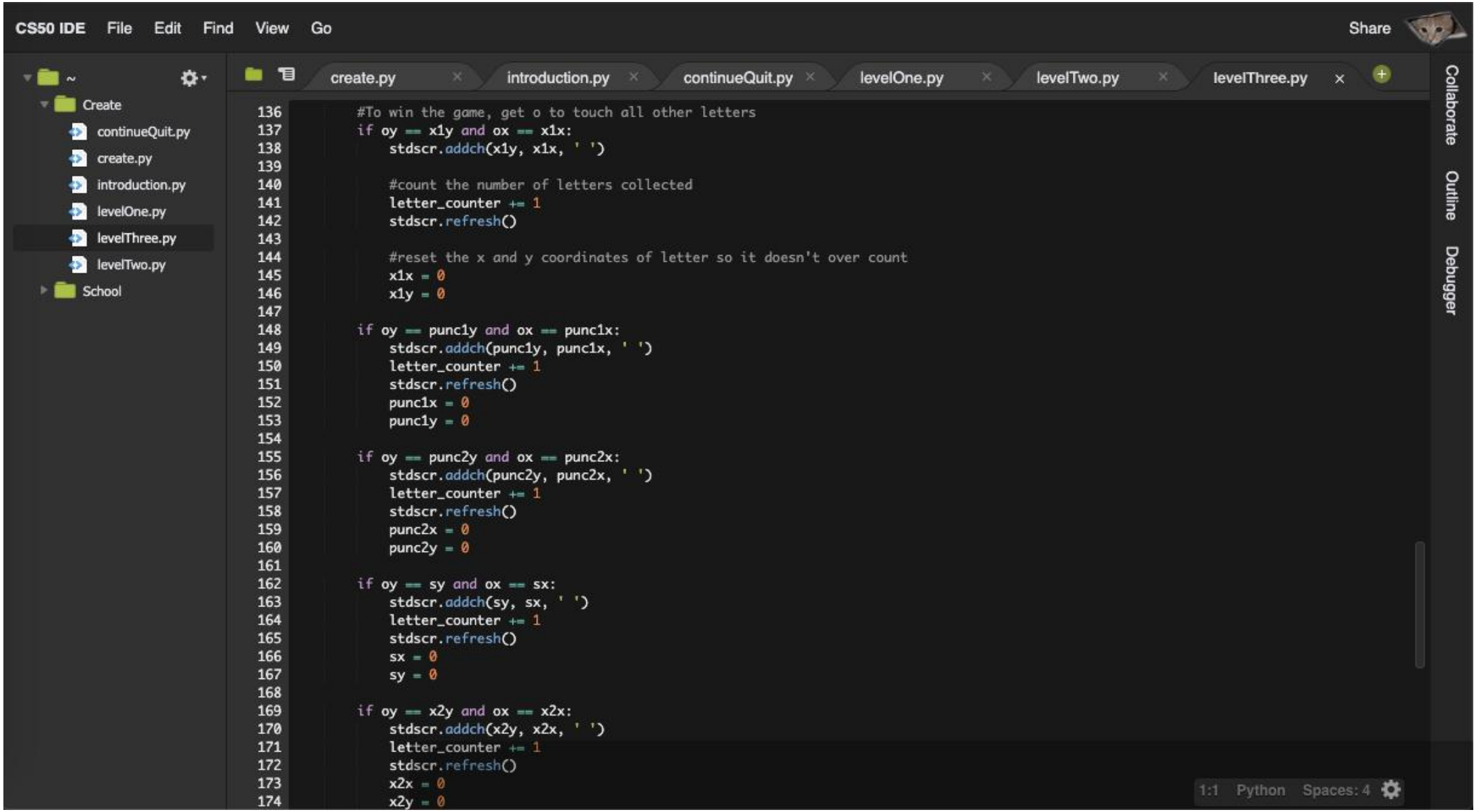

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
Collaborate
Outline
Debugger
154
155 if oy == wy and ox == wx:
156 stdscr.addch(wy, wx, ' ')
157 letter_counter += 1
158 stdscr.refresh()
159 wx = 0
160 wy = 0
161 if letter_counter >= 6:
162 #Passed level two
163 stdscr.clear()
164 stdscr.refresh()
165 break
166
167 #asks for the next move from user
168 stdscr.refresh()
169 user_input = stdscr.getch()
170
171 #end program
172 curses.endwin()
173
44:30
Python
Spaces: 4

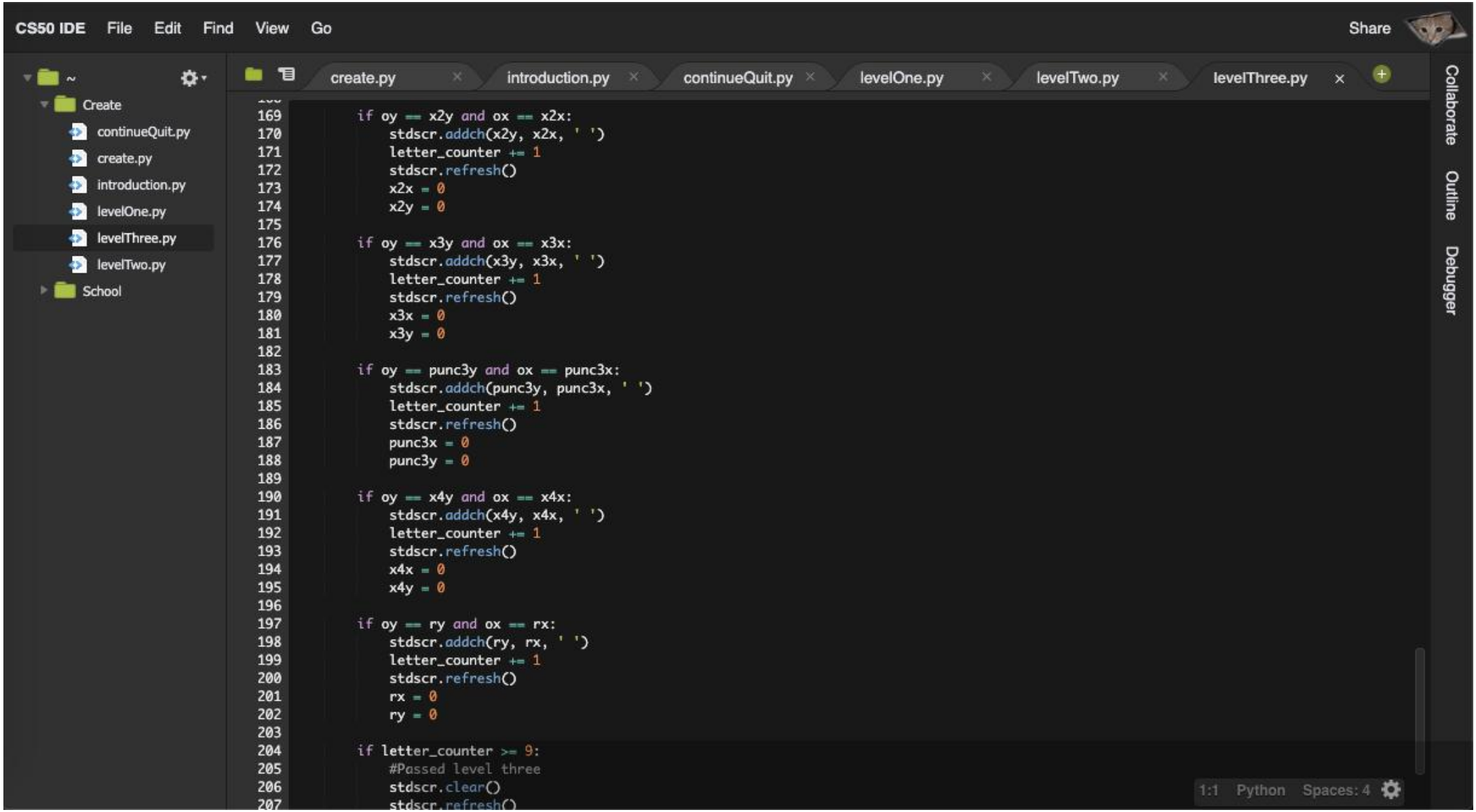

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
levelThree.py
Collaborate
Outline
Debugger
1 import curses
2 import time
3 import sys
4
5 import continueQuit
6
7 def three():
8 print("LEVEL THREE")
9 print("FINAL ROUND")
10 print("")
11 print("Collect all 9 letters to win the game!")
12 print("4/9 of the letters are lower-case 'x'. 3/9 are punctuation marks.")
13 print("Ready? GO!")
14 print("")
15 continueQuit.contQuit()
16
17 #initializes a screen, stored in variable stdscr
18 stdscr = curses.initscr()
19
20 #prompts that put curses into the appropriate operating settings
21 curses.noecho()
22 curses.cbreak()
23 stdscr.keypad(True)
24
25 #initializes a variable to count the number of letters collected
26 letter_counter = 0
27 stdscr.refresh()
28
29 #add an x to the screen
30 stdscr.addstr(1, 1, "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
31 stdscr.addstr(2, 1, "Xo X Xx X XXX XX X XX XX")
32 stdscr.addstr(3, 1, "XX X X X X X.XX XX X X X X X XX")
33 stdscr.addstr(4, 1, "X X XX X X XX X X X X XX X X X X X")
34 stdscr.addstr(5, 1, "X X X X X X XX X XX/XsXX X X")
35 stdscr.addstr(6, 1, "X XX XX X X X X X X XX X X")
36 stdscr.addstr(7, 1, "X X X XXX XxX X X X XX X X X X X")
37 stdscr.addstr(8, 1, "X X X XX XX X XxX X XX")
38 stdscr.addstr(9, 1, "XX XX X XX X X X XX XX X XX X~X")
39 stdscr.addstr(10, 1,"X X X X X X X X X X X X")
1:1 Python Spaces: 4

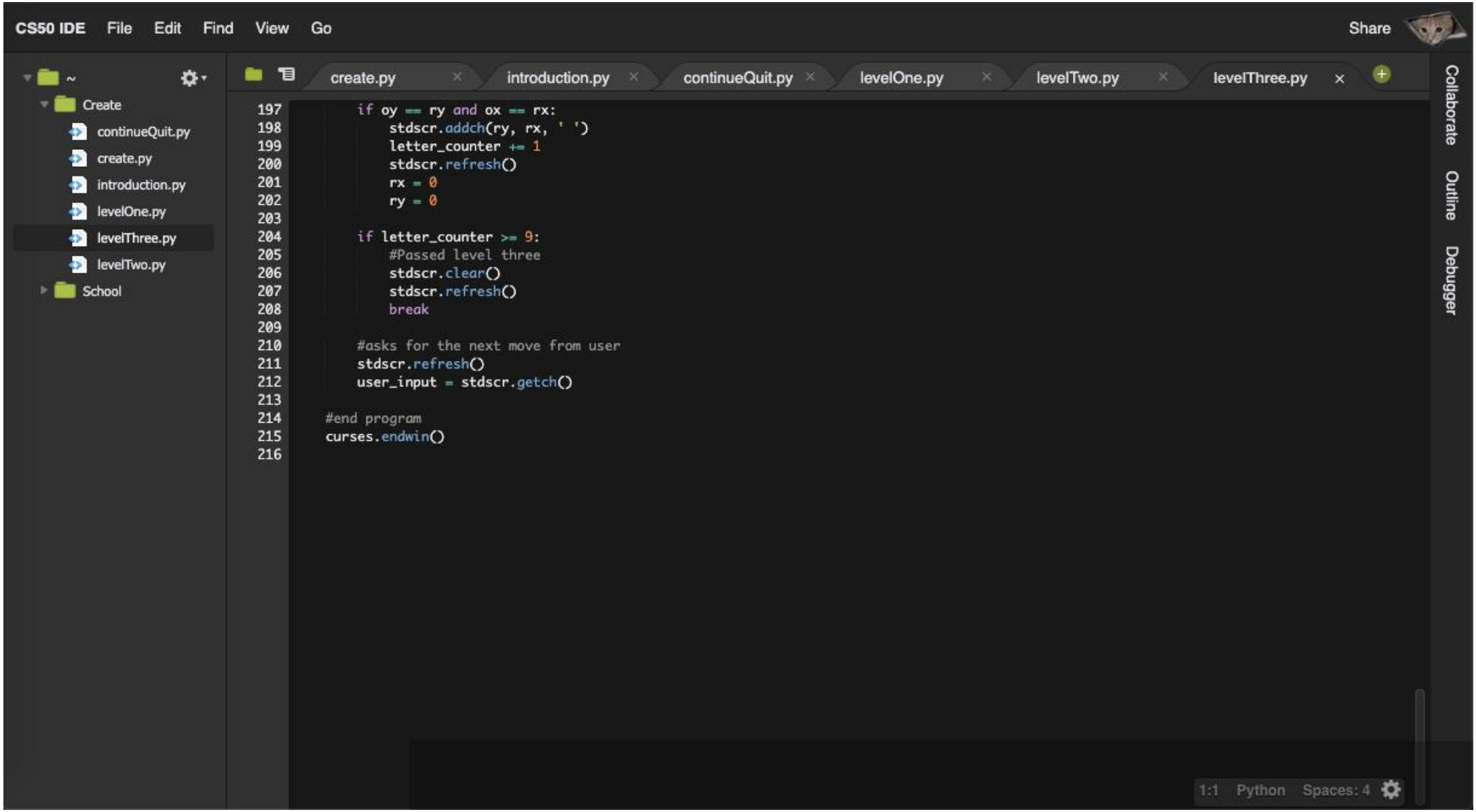

CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
levelThree.py
Collaborate
Outline
Debugger
28
29 #add an x to the screen
30 stdscr.addstr(1, 1, "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
31 stdscr.addstr(2, 1, "Xo X Xx X XXX XX X XX XX")
32 stdscr.addstr(3, 1, "XX X X X X X.XX XX X X X X X XX")
33 stdscr.addstr(4, 1, "X X XX X X XX X X X X XX X X X X")
34 stdscr.addstr(5, 1, "X X X X X X XX X XX/XsXX X X")
35 stdscr.addstr(6, 1, "X XX XX X X X X X X XX X X X")
36 stdscr.addstr(7, 1, "X X X XXX XxX X X X XX X X X X X")
37 stdscr.addstr(8, 1, "X X X XX XX X XxX X XX")
38 stdscr.addstr(9, 1, "XX XX X XX X X X XX XX X XX X~X")
39 stdscr.addstr(10, 1,"X X X X X X X X X X X X")
40 stdscr.addstr(11, 1,"X X X X XX X X Xx XX Xr XX X X")
41 stdscr.addstr(12, 1,"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
42
43 stdscr.refresh()
44
45 #x- and y- coordinates of o & y
46 ox = 2
47 oy = 2
48 stdscr.addch(oy, ox, 'o')
49
50
51 #x- and y- coordinates of each letter
52 x1x = 11
53 x1y = 2
54 stdscr.addch(x1y, x1x, 'x')
55 stdscr.refresh()
56
57 punc1x = 19
58 punc1y = 3
59 stdscr.addch(punc1y, punc1x, '.')
60 stdscr.refresh()
61
62 punc2x = 36
63 punc2y = 5
64 stdscr.addch(punc2y, punc2x, '/')
65 stdscr.refresh()
66
1:1 Python Spaces: 4


CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
levelThree.py
Collaborate
Outline
Debugger
67 sx = 38
68 sy = 5
69 stdscr.addch(sy, sx, 's')
70 stdscr.refresh()
71
72 x2x = 17
73 x2y = 7
74 stdscr.addch(x2y, x2x, 'x')
75 stdscr.refresh()
76
77 x3x = 24
78 x3y = 8
79 stdscr.addch(x3y, x3x, 'x')
80 stdscr.refresh()
81
82 punc3x = 43
83 punc3y = 9
84 stdscr.addch(punc3y, punc3x, '~')
85 stdscr.refresh()
86
87 x4x = 24
88 x4y = 11
89 stdscr.addch(x4y, x4x, 'x')
90 stdscr.refresh()
91
92 rx = 33
93 ry = 11
94 stdscr.addch(ry, rx, 'r')
95 stdscr.refresh()
96
97 #start the maze
98 user_input = stdscr.getch()
99
100 if user_input == ord('q'):
101 sys.exit()
102 curses.endwin()
103
104 while user_input != ord('q'):
105
1:1 Python Spaces: 4


CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
levelThree.py
Collaborate
Outline
Debugger
while user_input != ord('q'):
#move down
if user_input == ord('s') and chr(stdscr.inch(oy + 1, ox)) != 'X' and oy + 1 < 45:
stdscr.addch(oy, ox, ' ')
oy += 1
stdscr.addch(oy, ox, 'o')
stdscr.refresh()
#move up
if user_input == ord('w') and chr(stdscr.inch(oy - 1, ox)) != 'X' and oy - 1 < 45:
stdscr.addch(oy, ox, ' ')
oy -= 1
stdscr.addch(oy, ox, 'o')
stdscr.refresh()
#move left
if user_input == ord('a') and chr(stdscr.inch(oy, ox - 1)) != 'X' and ox - 1 < 45:
stdscr.addch(oy, ox, ' ')
ox -= 1
stdscr.addch(oy, ox, 'o')
stdscr.refresh()
#move right
if user_input == ord('d') and chr(stdscr.inch(oy, ox + 1)) != 'X' and ox + 1 < 45:
stdscr.addch(oy, ox, ' ')
ox += 1
stdscr.addch(oy, ox, 'o')
stdscr.refresh()
#To win the game, get o to touch all other letters
if oy == x1y and ox == x1x:
stdscr.addch(x1y, x1x, ' ')
#count the number of letters collected
letter_counter += 1
stdscr.refresh()
1:1
Python
Spaces: 4


CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
levelThree.py
Collaborate
Outline
Debugger
#To win the game, get o to touch all other letters
if oy == x1y and ox == x1x:
stdscr.addch(x1y, x1x, ' ')
#count the number of letters collected
letter_counter += 1
stdscr.refresh()
#reset the x and y coordinates of letter so it doesn't over count
x1x = 0
x1y = 0
if oy == punc1y and ox == punc1x:
stdscr.addch(punc1y, punc1x, ' ')
letter_counter += 1
stdscr.refresh()
punc1x = 0
punc1y = 0
if oy == punc2y and ox == punc2x:
stdscr.addch(punc2y, punc2x, ' ')
letter_counter += 1
stdscr.refresh()
punc2x = 0
punc2y = 0
if oy == sy and ox == sx:
stdscr.addch(sy, sx, ' ')
letter_counter += 1
stdscr.refresh()
sx = 0
sy = 0
if oy == x2y and ox == x2x:
stdscr.addch(x2y, x2x, ' ')
letter_counter += 1
stdscr.refresh()
x2x = 0
x2y = 0
1:1
Python
Spaces: 4


CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
levelThree.py
Collaborate
Outline
Debugger
169 if oy == x2y and ox == x2x:
170 stdscr.addch(x2y, x2x, ' ')
171 letter_counter += 1
172 stdscr.refresh()
173 x2x = 0
174 x2y = 0
175
176 if oy == x3y and ox == x3x:
177 stdscr.addch(x3y, x3x, ' ')
178 letter_counter += 1
179 stdscr.refresh()
180 x3x = 0
181 x3y = 0
182
183 if oy == punc3y and ox == punc3x:
184 stdscr.addch(punc3y, punc3x, ' ')
185 letter_counter += 1
186 stdscr.refresh()
187 punc3x = 0
188 punc3y = 0
189
190 if oy == x4y and ox == x4x:
191 stdscr.addch(x4y, x4x, ' ')
192 letter_counter += 1
193 stdscr.refresh()
194 x4x = 0
195 x4y = 0
196
197 if oy == ry and ox == rx:
198 stdscr.addch(ry, rx, ' ')
199 letter_counter += 1
200 stdscr.refresh()
201 rx = 0
202 ry = 0
203
204 if letter_counter >= 9:
205 #Passed level three
206 stdscr.clear()
207 stdscr.refresh()
1:1
Python
Spaces: 4


CS50 IDE
File
Edit
Find
View
Go
Share
~
Create
continueQuit.py
create.py
introduction.py
levelOne.py
levelThree.py
levelTwo.py
School
create.py
introduction.py
continueQuit.py
levelOne.py
levelTwo.py
levelThree.py
Collaborate
Outline
Debugger
197 if oy == ry and ox == rx:
198 stdscr.addch(ry, rx, ' ')
199 letter_counter += 1
200 stdscr.refresh()
201 rx = 0
202 ry = 0
203
204 if letter_counter >= 9:
205 #Passed level three
206 stdscr.clear()
207 stdscr.refresh()
208 break
209
210 #asks for the next move from user
211 stdscr.refresh()
212 user_input = stdscr.getch()
213
214 #end program
215 curses.endwin()
216
1:1
Python
Spaces: 4