

**Lucrare pentru obținerea
atestatului profesional la informatică**

Tetrísmania

Profesor coordonator
Filip Adela

Candidat
Diaconescu Răzvan

Cuprins

INTRODUCERE	1
PYCHARM.....	2
Prezentare aplicației	3
I. Logica jocului	5
II. Scoreboard.....	7
Codul sursă.....	8
Bibliografie	15

INTRODUCERE

TETRISMANIA este o aplicație de calculator care poate fi folosită pentru a se juca o iterație de Tetris proiectată de mine. Motivul pentru care am ales această temă a fost pentru a învăța un limbaj nou de programare, acest limbaj fiind `Python`, și pentru a învăța conceptele utilizate în programarea orientată obiect. Acest tip de programare este o parte importantă a construirii aplicațiilor și software-ului.

Aplicația TETRISMANIA este realizată cu ajutorul mediului PyCharm.

PyCharm este un IDE (Integrated Development Environment) care folosește limbajul `Python` pentru a programa. Acest software oferă un asistent și analizator de cod care marchează sintaxa și erorile și abilitatea de naviga în cod și proiect cu abilitatea de a vedea structura fișierelor și abilitatea de a folosi clase.

PyCharm are două ediții diferite: *Professional* și *Community*. Versiunea *Community* este open-source, dar are mai puține dotări în comparație cu *Professional*. În acest proiect am folosit versiunea *Community*.

PyCharm poate fi rulat pe Windows, macOS și Linux.



PYCHARM

PyCharm este o aplicație Windows cu un GUI inspirat de aplicația Codeblocks. Singura diferență dintre cele două aplicații fiind mediile de aplicație pe care le pot folosi aplicațiile: pentru PyCharm Python, și pentru Codeblocks C++, C și JAVA.

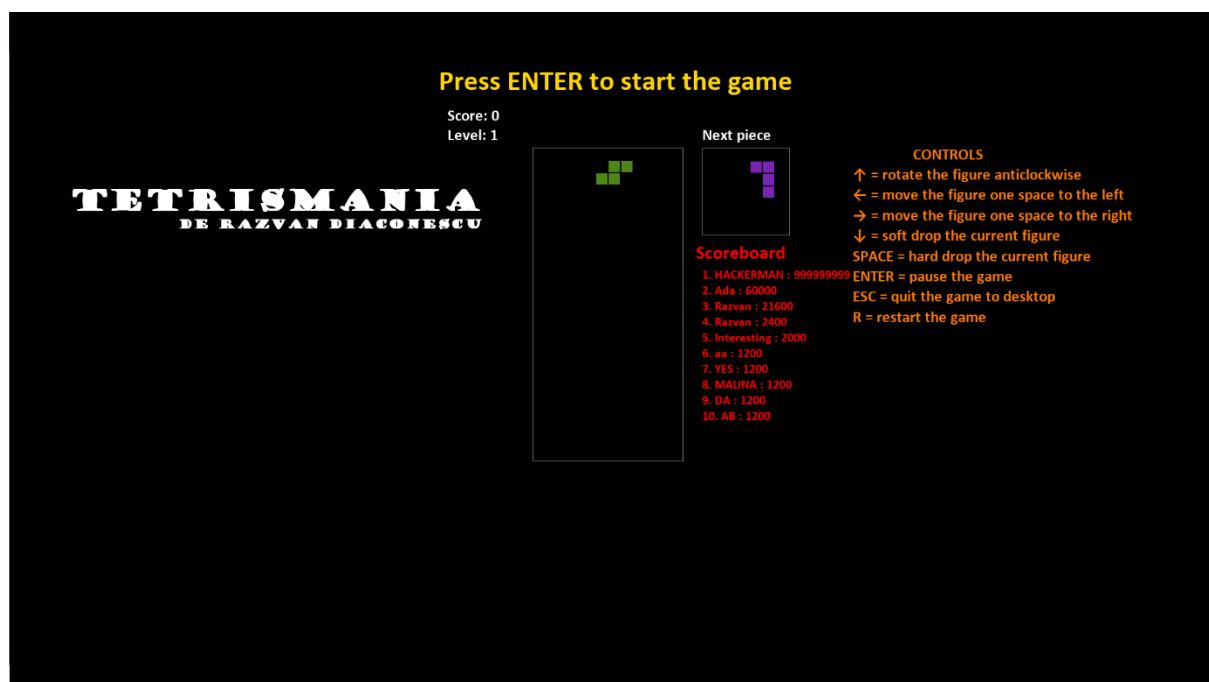


- Window header conține o mulțime de widget-uri care dau access la cele mai populare acțiuni: project widget, VCS widget și run widget. De asemenea îți oferă abilitatea de a deschide Code With Me, Search Everywhere și Settings.
- Un Project tool window pe partea stângă a ecranului unde afișează fișierele proiectului.
- Un Editor pe partea dreapta, unde se scrie codul. Deasupra editorului sunt tab-uri ce fac navigarea între fișierele deschise mai ușor.
- Meniurile de context se deschid când se face click dreapta pe un element al interfeței sau o parte de cod și se va afișa toate acțiunile ce se pot executa.
- Un bar de navigație ce permite navigarea rapidă printre fișierele și documentele proiectului.
- Gutter, linia vertical de lângă editor, arată punctele de întrerupere pe care le are codul și oferă o metodă de navigare prin cod ierarhic cum ar fi la definiții sau declarații. De asemenea arată și numărul liniei.
- Fereastra de unelte sunt ferestre specializate și atasate în subsol și laterar spațiului de lucru. Aceste oferă acces la project management, căutare în codul sursă și navigație, rulare, testare și debugging.
- Bara de status indică statusul proiectului și a IDE-ului, pe lângă acestea mai arată și mesaje de avertizare sau informare despre file encoding, separarea liniilor, profil de inspecție și multe altele. Acesta mai oferă și o metodă rapidă de a accesa setările pentru interpretatorul de Python.

Prezentare aplicației

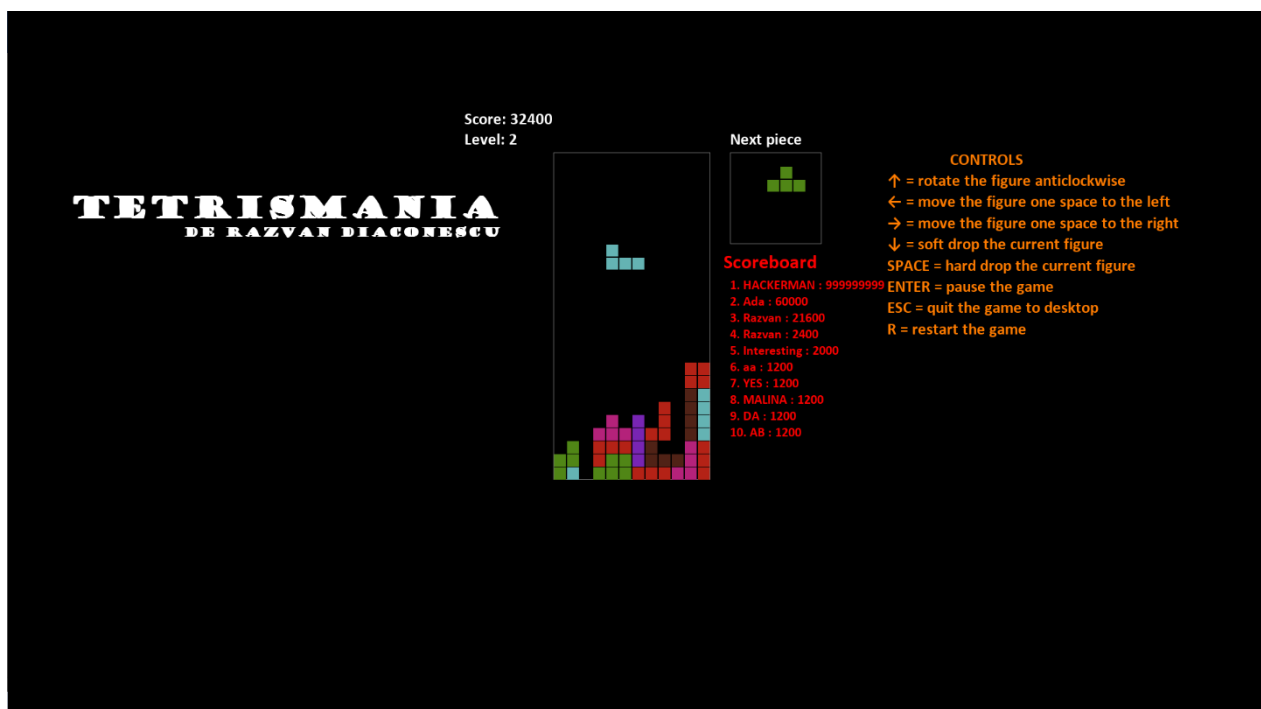
Jocul va rula cu ajutorul unei structuri repetitive ce se va opri doar în cazul în care `done` are valoarea 1. Această structură repetitivă va conține:

- Limbajul de programare `Python` are acces la o bibliotecă numită `pygame`, ce conține câteva funcții și clase ce ajută la proiectarea unui joc, ne vom folosi de `pygame.event.get()` pentru a obține un vector de tastele apășate. Vom parcurge acest vector cu o structură repetitivă liniară, unde vom numi event pointer-ul care indică poziția curentă a elementului din `pygame.event.get()`. În structura repetitivă vom verifica ce tastă este apăsată și apelăm comenzile corespundătoare mișcării cerute sau în cazul salvării scorului vom memora ce nume va scrie jucătorul. Comenzile posibile vor fi:
 - ✓ Săgeată sus pentru a roti figura
 - ✓ Săgeată dreapta sau stânga mută figura cu un spațiu în direcția respective
 - ✓ Săgeată jos face figura să cadă mai rapid
 - ✓ Spacebar va pune instant figura unde ar cădea în poziția curentă
 - ✓ Enter va pune jocul pe pauză
 - ✓ ESC va închide aplicația
 - ✓ Tasta R va restarta jocul



- Grafica jocului este realizată de asemenea cu ajutorul bibliotecii pygame, ne vom folosi de `pygame.draw.rect` pentru a proiecta figurile, spațiul de joc și de sistemul de font pe care îl conține pygame pentru a afișa text pe ecran.
- ✓ `pygame.Rect(numărul întreg x, numărul întreg y, numărul întreg l, numărul întreg h)` proiectează un dreptunghi de lungime `l` și înălțime `h`, colțul din stânga sus fiind situat în `x,y`. `x, y, l` și `h` sunt măsurate în pixeli.
- ✓ `pygame.font` este folosit pentru a proiecta text cu diferite font-uri și mărimi pe ecran. Primul pas pentru a afișa un text pe ecran este crearea unui font cu `pygame.font.SysFont`. După ce am creat un font putem folosi `pygame.render` pentru a obține un obiect în care este memorat un string și culoarea în care dorim să afișăm textul. În final vom folosi `screen.blit` pentru a afișa pe ecran în poziția de coordonate `x` și `y`.
- În funcție de scorul curent jocul își va putea crește dificultatea prin accelerarea vitezei cu care piesele coboară.

TETRISMANIA folosește programarea orientată obiect pentru a memora datele pentru scoreboard, figurile din joc și jocul în sine.

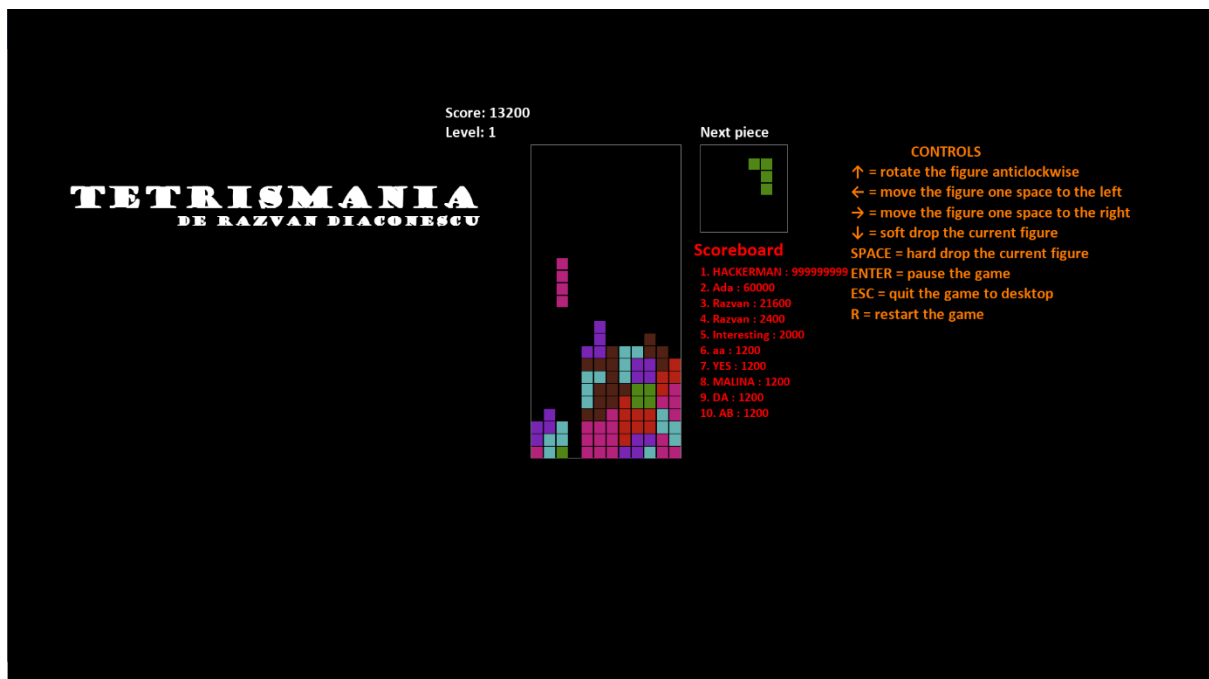


I. Logica jocului

- Clasa Figure este folosită pentru a reprezenta figurile din joc.
 - ✓ În două variabile `x` și `y` vom ține poziția în spațiul de joc al figurii, în variabila `type` vom memora poziția pe care o are forma obiectului ales în vectorul global de forme de obiecte, în `color` vom memora poziția culorii pe care o va avea obiectul dintr-un vector global de culori.
 - ✓ Vom folosi 3 definiții în acest obiect, una pentru inițializare obiectului, una pentru returnarea pozițiilor pe care le ocupa figura și una pentru returnarea pozițiilor ocupate în cazul în care figura este rotită.
 - ✓ La inițializarea unui obiect vom cere două variabile `x` și `y` unde vom salva coordonatele obiectului, în variabila `type` vom pune cu ajutorul bibliotecii `random` vom alege un indice aleatoriu care să fie situat în vectorul `figures`, vom folosi aceeași metodă și pentru `color` dar vom alege un indice din vectorul de culori.
 - ✓ Definiția `image` va returna imaginea figurii.
 - ✓ Definiția `rotate` va schimba valoarea variabilei `rotation` adăugând 1 și aplicând modulo lungimea figurilor de poziția `type`.

- Clasa Tetris va conține datele folosite pentru rularea jocului.
 - ✓ Variabile folosite sunt: `level` care menține nivelul curent al jocului care va dicta viteza cu care vor pica piesele, variabila `state` unde vom memora statusul jocului, două variabile `height` și `width` ce vor reprezenta lungimea și lățimea matricei `field` pe care funcționează jocul, `score` ce va memora punctajul curent al runde curente, două obiecte de tip `Figure` numite `figure` și `next_figure` ce vor reprezenta figura curentă și figura ce va urma.
 - ✓ În definiția de inițializare vom goli matricea de joc și atribuim variabilelor `score`, `level` și `state` valorile de default.
 - ✓ Definiția `new_figure` va genera un obiect de tip `Figure` și o atribuie variabilei `figure`.
 - ✓ `intersects` va verifica dacă piesa curentă se intersectează cu un spațiu ocupat din `fields`. Se va returna `true` dacă se găsește un spațiu ocupat, `false` în caz contrar.
 - ✓ `break_lines` va distruge liniile complet ocupat și va muta liniile de deasupra cu un spațiu mai jos.

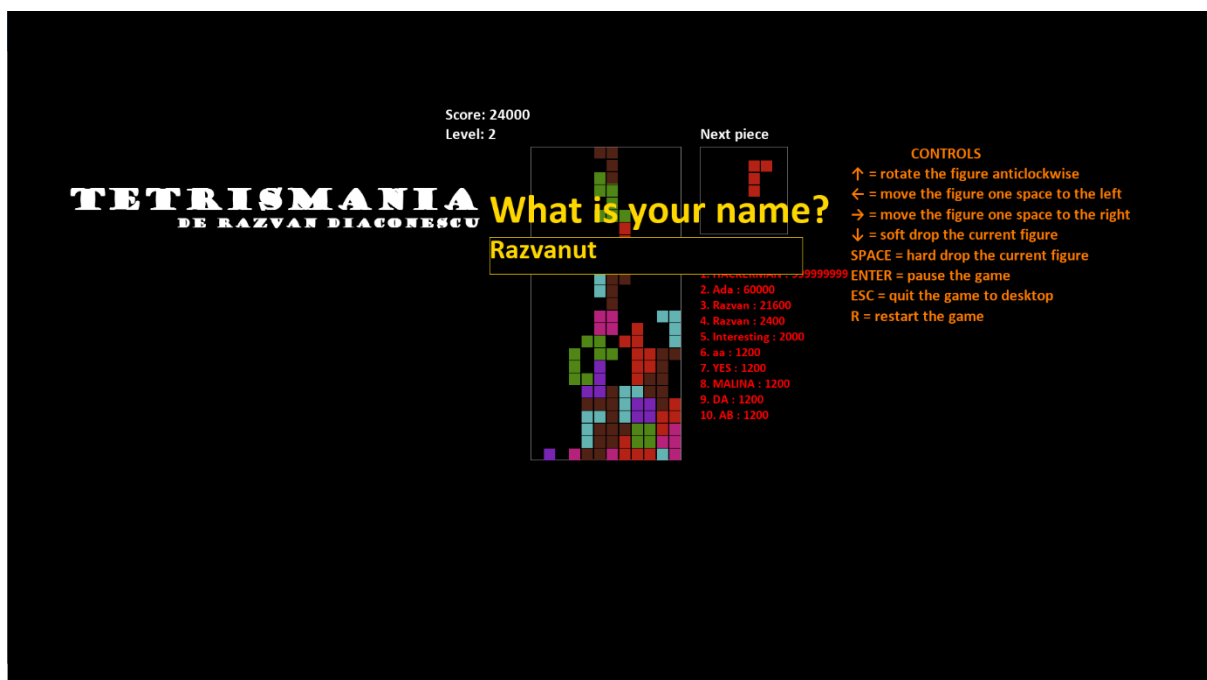
- ✓ `go_space` va folosi o structură repetitivă unde cât timp `not intersects` este adevărată figura curentă se va muta cu un spațiu în jos, când ieșim din structura repetitivă, vom muta figura cu un spațiu în sus.
- ✓ Definiția `go_side` va primi o valoare `dx` ce va fi fie `-1` sau `1`, figura curentă o memorăm într-o variabilă `old_x`, modificăm coordonata `x` a lui `figure` și dacă `intersects` returnează `true`, vom egala `figure` cu `old_x`.
- ✓ `Rotate` va salva în `old_rotation` valoarea `rotation` a lui `figure`, apelăm `figure.rotate()` și dacă `intersects` returnează `true` vom atribui lui `figure.rotation` `old_rotation`.



II. Scoreboard

Tabelul cu cele mai mari scoruri se folosește de o clasă numită `PlayerData`, aceasta clasă conține:

- Un string care o vom folosi pentru a memora numele jucătorului, inițial ea va avea valoare de default “dummy” și o variabilă de tip `int` ce va conține scorul jucătorului, inițializată cu valoarea 0.
- Pe lângă aceste date vom avea o definiție ce va compara două elemente de tip `PlayerData`.
- Pentru funcționarea tabelului de scor vom scrie și citi într-un document cu numele “scoreboard.txt”, în cazul în care acest fișier nu există vom crea unul cu acest nume.
- Înainte de a începe jocul vom citi toate datele din fișierul “scoreboard.txt” și le vom memora într-un container global cu numele `scoreboard_data`. Când jucătorul va pierde, jocul îi va cere să introducă un nume. După aceasta vom crea un obiect de tip `PlayerData` ce va conține acest nume și scorul jucătorului și acest obiect îl vom introduce în `scoreboard_data`.
- Vom sorta `scoreboard_data` și vom șterge din `scoreboard.txt` toate caracterele și vom scrie datele din `scoreboard_data`.



Codul sursă

```

1 import pygame
2 import random
3 import os.path
4 from dataclasses import dataclass
5 colors = [
6     (0, 0, 0),
7     (120, 179, 179),
8     (100, 179, 179),
9     (80, 34, 22),
10    (80, 134, 22),
11    (180, 34, 22),
12    (180, 34, 122),
13 ]
14 figures = [
15     [[1, 5, 9, 10], [4, 5, 6, 7]],
16     [[4, 5, 9, 10], [2, 6, 5, 9]],
17     [[6, 7, 9, 10], [1, 5, 6, 10]],
18     [[1, 2, 5, 9], [9, 4, 5, 6], [1, 5, 9, 8], [4, 5, 6, 10]],
19     [[1, 2, 6, 10], [5, 6, 7, 9], [2, 6, 10, 11], [5, 9, 6, 7]],
20     [[1, 4, 5, 9], [1, 4, 5, 9], [4, 5, 6, 9], [1, 5, 6, 9]],
21     [[1, 2, 5, 6]],
22 ]
23 gsx=25
24 gsy=12
25 resolution=1920
26 resolutiony=1080
27 @dataclass
28 class PlayerData:
29     playerName:str="dummy"
30     playerscore:int=0

```

Run: tetris.py
 C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
 pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
 Hello from the pygame community. <https://www.pygame.org/contribute.html>
 True
 Process finished with exit code 0

```

31 def __init__(self, other):
32     if self.playerscore!=other.playerscore:
33         return self.playerscore=other.playerscore
34     else: return self.playername=other.playername
35 def chartoint(x):
36     return {
37         '0':0,
38         '1':1,
39         '2':2,
40         '3':3,
41         '4':4,
42         '5':5,
43         '6':6,
44         '7':7,
45         '8':8,
46         '9':9
47     }[x]
48 class Figure:
49     x = 0
50     y = 0
51     def __init__(self, x, y):
52         self.x = x
53         self.y = y
54         self.type = random.randint(0, len(figures) - 1)
55         self.color = random.randint(1, len(colors) - 1)
56         self.rotation = 0
57     def image(self):
58         return figures[self.type][self.rotation]

```

Run: tetris.py
 C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
 pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
 Hello from the pygame community. <https://www.pygame.org/contribute.html>
 True
 Process finished with exit code 0

```

11 def rotate(self):
12     self.rotation = (self.rotation + 1) % len(figures[self.type])
13
14 class Tetris:
15     level = 1
16     score = 0
17     state = "startscreen"
18     field = []
19     height = 0
20     width = 0
21     x = resolutionx/2.3
22     y = resolutiony/5
23     xx = resolutionx/2*1.125
24     yy = resolutiony/5*1.1
25     zoom = 20
26     figure = None
27     next_figure = None
28
29     def __init__(self, height, width):
30         self.height = height
31         self.width = width
32         self.field = []
33         self.score = 0
34         self.level = 1
35         self.state = "startscreen"
36         for i in range(height):
37             new_line = []
38             for j in range(width):
39                 new_line.append(0)
40             self.field.append(new_line)

```

Run: tetris.py

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atetstat\tetris.py
 pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
 Hello from the pygame community. <https://www.pygame.org/contribute.html>
 True
 Process finished with exit code 0

```

41 def new_figure(self):
42     self.figure = Figure(4,0)
43
44 def new_next_figure(self):
45     self.next_figure = Figure(4,0)
46
47 def swap_figure(self):
48     max = self.figure
49     self.figure = self.next_figure
50     self.next_figure = max
51
52 def intersects(self):
53     intersection = False
54     for i in range(4):
55         for j in range(4):
56             if i + self.figure.y > self.height - 1 or \
57                 j + self.figure.x > self.width - 1 or \
58                 j + self.figure.x < 0 or \
59                 self.field[i + self.figure.y][j + self.figure.x] > 0:
60                 intersection = True
61     return intersection
62
63 def break_lines(self):
64     lines = 0
65     for i in range(1, self.height):
66         zeros = 0
67         for j in range(self.width):
68             if self.field[i][j] == 0:
69                 zeros += 1
70     if zeros == 0:

```

Run: tetris.py

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atetstat\tetris.py
 pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
 Hello from the pygame community. <https://www.pygame.org/contribute.html>
 True
 Process finished with exit code 0

```

121         lines += 1
122         for i1 in range(1, 1, -1):
123             for j in range(self.width):
124                 self.field[i1][j] = self.field[i1 - 1][j]
125         self.score += lines * 2 * 1000
126
127     def go_space(self):
128         while not self.intersects():
129             self.figure.y += 1
130             self.figure.y -= 1
131             self.freeze()
132
133     def go_down(self):
134         self.figure.y += 1
135         if self.intersects():
136             self.figure.y -= 1
137             self.freeze()
138
139     def freeze(self):
140         for i in range(4):
141             for j in range(4):
142                 if 1 <= i <= 4 & j in self.figure.image():
143                     self.field[i + self.figure.y][j] + self.figure.x = self.figure.color
144                 self.break_lines()
145                 self.swap_figure()
146                 self.new_next_figure()
147                 if self.intersects():
148                     self.state = "recordscore"
149
150     def go_side(self, dx):
151         self.swap_figure()

```

Run: C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. <https://www.pygame.org/contribute.html>
True
Process finished with exit code 0

```

151         self.swap_figure()
152
153     old_x = self.figure.x
154     self.figure.x += dx
155     if self.intersects():
156         self.figure.x = old_x
157
158     def rotate(self):
159         old_rotation = self.figure.rotation
160         self.figure.rotate()
161         if self.intersects():
162             self.figure.rotation = old_rotation
163
164     pygame.init()
165
166     # colors
167     BLACK = (0, 0, 0)
168     WHITE = (255, 255, 255)
169     GRAY = (128, 128, 128)
170     BLUE = (0, 0, 255)
171     BRONZE = (205, 127, 50)
172     SILVER = (192, 192, 192)
173     GOLD = (255, 215, 0)
174     RED = (255, 0, 0)
175     size = (resolutions, resolutions)
176     screen = pygame.display.set_mode(size)
177
178     pygame.display.set_caption("Tetris")
179
180     done = False
181     clock = pygame.time.Clock()
182     print(clock)
183     fps = 15

```

Run: C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. <https://www.pygame.org/contribute.html>
True
Process finished with exit code 0

```

181
182
183 difficulty = [13, 11, 7, 5, 3, 2]
184 pressing_down = False
185 game = tetris(gsx, gsy)
186 counter = 0
187
188 if file_exists=os.path.exists("scoreboard.txt"):
189     if (file_exists==False):
190         scoreboard=open("scoreboard.txt", "w")
191         scoreboard.close()
192     print (file_exists)
193     with open("scoreboard.txt", "r+") as file:
194         scoreboard=iter(file.read().split())
195         #scoreboard=tetris(scoreboard.txt, "r+")
196         #scoreboard=iter(scoreboard.txt.split())
197         counter=0
198         quitgame=0
199         scoreboard_data=[PlayerData() for i in range(10)]
200         playername=""
201         input_rect=pygame.Rect(resolution/3+1.4, resolution/2, 200, 50)
202         #font
203         titlefont=pygame.font.SysFont('goudy stout', 45, True, False)
204         authorfont=pygame.font.SysFont('goudy stout', 20, True, False)
205         #print (pygame.font.get_fonts())
206         fontbegin=pygame.font.SysFont('calibri', 45, True, False)
207         startfont=pygame.font.SysFont('calibri', 45, True, False)
208         scoreboardfont = pygame.font.SysFont('calibri', 20, True, False)
209         B[scoreboardfont=pygame.font.SysFont('calibri', 20, True, False)
210         font = pygame.font.SysFont('calibri', 25, True, False)
211         font1 = pygame.font.SysFont('calibri', 45, True, False)
212         swap_figure()

```

Run: tetris.py

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. <https://www.pygame.org/contribute.html>
True
Process finished with exit code 0

```

211
212 pausefont=pygame.font.SysFont('calibri', 45, True, False)
213 while True:
214     try:
215         scoreboard_data[counter].playername = next(scoreboard)
216         number=next(scoreboard)
217         #print
218         scoreboard_data[counter].playerscore=0
219         for char in number:
220             if (char>='0' and char<='9'):
221                 scoreboard_data[counter].playerscore+=int(char)
222             #p+=10
223         #scoreboard_data[counter].playerscore=int = next(scoreboard)
224         #print(scoreboard_data[counter].playername, scoreboard_data[counter].playerscore)
225         #print(scoreboard_data[1].playername, scoreboard_data[1].playerscore, counter)
226         counter = counter + 1
227     except StopIteration:
228         break
229 if (game.state=="startscreen"):
230     while not done:
231         #print(game.state)
232         screen.fill(BLACK)
233         if game.figure is None:
234             game.new_figure()
235             game.swap_figure()
236             game.new_next_figure()
237         #print(counter)
238         #print (game.figure.type, game.figure.color, game.next_figure.type, game.next_figure.color)
239         counter += 1
240         if counter > 100000:
241             counter = 0

```

Run: tetris.py

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. <https://www.pygame.org/contribute.html>
True
Process finished with exit code 0

```

241 if counter % (difficulty(game.level)) == 0 or pressing_down:
242     if game.state == "start":
243         game.go_down()
244     for event in pygame.event.get():
245         if event.type == pygame.QUIT:
246             done = True
247         if event.type == pygame.KEYDOWN:
248             if (event.key == pygame.K_RETURN and (game.state=="startscreen" or game.state=="break")):
249                 game.state="start"
250             elif (event.key == pygame.K_RETURN and game.state=="start"):
251                 game.state="break"
252             elif (event.key == pygame.K_ESCAPE and (game.state=="break" or game.state=="startscreen")):
253                 quitgame()
254             elif (event.key == pygame.K_r and game.state=="recordscore"):
255                 game.figure=None
256                 game.next.figure=None
257                 game._init_((gx, gvy))
258                 game.state="start"
259             elif event.key == pygame.K_h and game.state == "start":
260                 # game.show_figure()
261                 # print (game.intersects())
262                 # if game.intersects() is True:
263                     # game.show_figure()
264                     # print(True)
265             if event.key == pygame.K_UP and game.state == "start":
266                 game.rotate()
267             if event.key == pygame.K_DOWN and game.state=="start":
268                 pressing_down = True
269             if event.key == pygame.K_LEFT and game.state=="start":
270                 game.go_side(-1)

```

Run console output:

```

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atetstat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. https://www.pygame.org/contribute.html
True
Process finished with exit code 0

```

```

271 if event.key == pygame.K_RIGHT and game.state=="start":
272     game.go_side(1)
273 if event.key == pygame.K_SPACE and game.state=="start":
274     game.go_space()
275 if game.state=="recordscore":
276     if (event.type == pygame.KEYDOWN):
277         if event.key == pygame.K_BACKSPACE:
278             playername = playername[:-1]
279         else:
280             playername += event.unicode
281     if event.type==pygame.KEYDOWN and event.key==pygame.K_RETURN:
282         if playername == "":
283             playername = "dummy"
284         neoscontender = PlayerData(playername, game.score)
285         game.state="gameover"
286         playername=""
287 if game.state == "gameover" and (event.key == pygame.K_SPACE or event.key==pygame.K_RETURN):
288     # aici trebuie salvata scorurile
289     if 0
290     scoreboard_data[is]:neoscontender
291     scoreboard_data.sort()
292     scoreboard=open("scoreboard.txt","r+")
293     scoreboard.truncate(0)
294     scoreboard=open("scoreboard.txt","a")
295     for it in range(10):
296         scoreboard.write(str(scoreboard_data[it].playername)+" "+str(scoreboard_data[it].playerscore)+" ")
297     scoreboard.close()
298     game._init_((gx, gvy))
299 if event.key == pygame.K_ESCAPE:

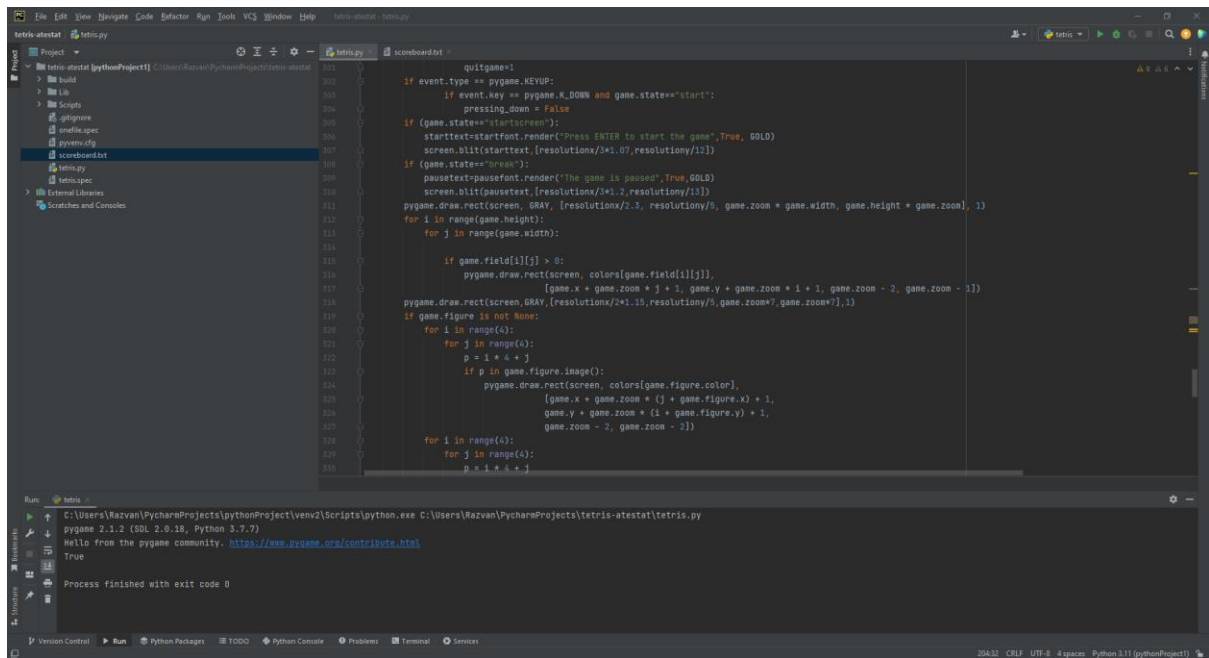
```

Run console output:

```

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atetstat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. https://www.pygame.org/contribute.html
True
Process finished with exit code 0

```



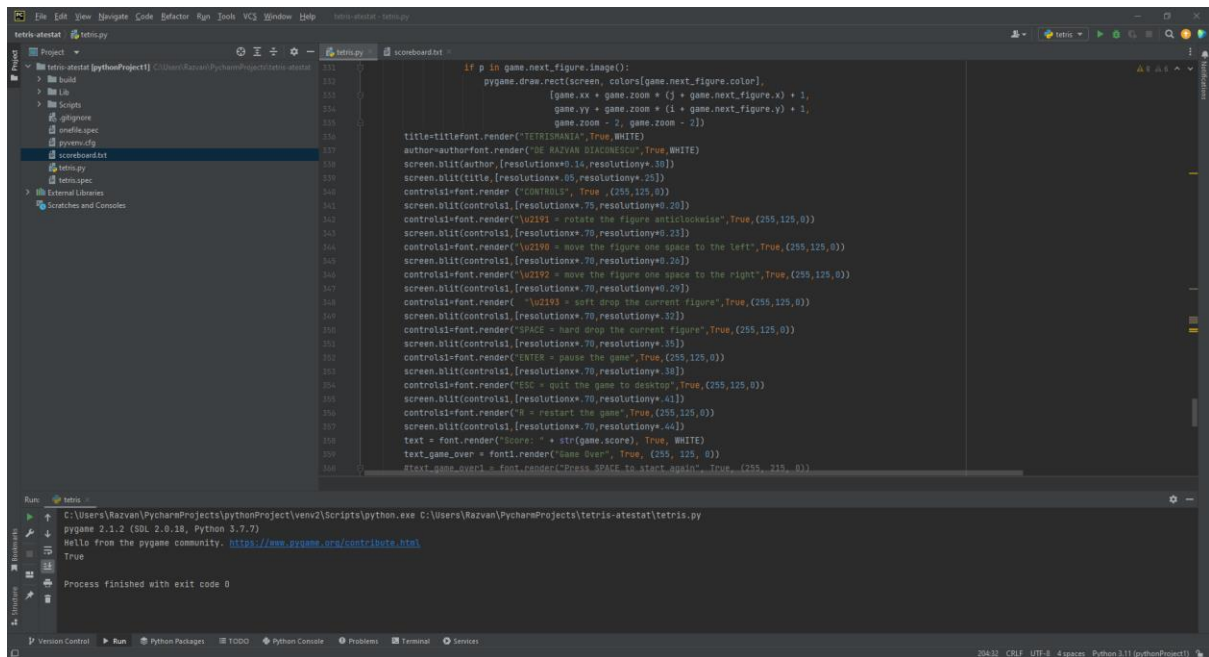
```

101 quitgame=1
102 if event.type == pygame.KEYUP:
103     if event.key == pygame.K_DOWN and game.state=="start":
104         pressing_down = False
105 if (game.state=="startscreen"):
106     starttext=startfont.render("Press ENTER to start the game",True, GOLD)
107     screen.blit(starttext,[resolutionx/4,0,resolutiony/12])
108 if (game.state=="break"):
109     pausetext=pausefont.render("The game is paused",True,GOLD)
110     screen.blit(pausetext,[resolutionx/4,2,resolutiony/13])
111 pygame.draw.rect(screen, GRAY, [resolutionx/2.3, resolutiony/5, game.zoom + game.width, game.height + game.zoom], 1)
112 for i in range(game.height):
113     for j in range(game.width):
114         if game.field[i][j] > 0:
115             pygame.draw.rect(screen, colors[game.field[i][j]],
116                             [game.x + game.zoom * j + 1, game.y + game.zoom * i + 1, game.zoom - 2, game.zoom - 1])
117 pygame.draw.rect(screen, GRAY, [resolutionx/2*1.15, resolutiony/5, game.zoom*7, game.zoom*7], 1)
118 if game.figure is not None:
119     for i in range(4):
120         for j in range(4):
121             p = i + 4 * j
122             if p in game.figure.image():
123                 pygame.draw.rect(screen, colors[game.figure.color],
124                                 [game.x + game.zoom * (j + game.figure.x) + 1,
125                                 game.y + game.zoom * (i + game.figure.y) + 1,
126                                 game.zoom - 2, game.zoom - 2])
127         for j in range(4):
128             p = i + 4 * j
129
130
131

```

Run

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. <https://www.pygame.org/contribute.html>
True
Process finished with exit code 0



```

131 if p in game.next_figure.image():
132     pygame.draw.rect(screen, colors[game.next_figure.color],
133                     [game.xx + game.zoom * (j + game.next_figure.x) + 1,
134                     game.yy + game.zoom * (i + game.next_figure.y) + 1,
135                     game.zoom - 2, game.zoom - 2])
136
137 title=titlefont.render("TETRISMANIA",True,WHITE)
138 author=authorfont.render("© RAZVAN DUMBESCU",True,WHITE)
139 screen.blit(author,[resolutionx*0.16,resolutiony*10])
140 screen.blit(title,[resolutionx*0.05,resolutiony*25])
141 controls=font.render("CONTROLS",True,(255,125,0))
142 screen.blit(controls1,[resolutionx*75,resolutiony*0.20])
143 controls=font.render("\u2191 = rotate the figure anticlockwise",True,(255,125,0))
144 screen.blit(controls1,[resolutionx*70,resolutiony*0.23])
145 controls=font.render("\u2190 = move the figure one space to the left",True,(255,125,0))
146 screen.blit(controls1,[resolutionx*70,resolutiony*0.26])
147 controls=font.render("\u2192 = move the figure one space to the right",True,(255,125,0))
148 screen.blit(controls1,[resolutionx*70,resolutiony*0.29])
149 controls=font.render(" \u2193 = soft drop the current figure",True,(255,125,0))
150 screen.blit(controls1,[resolutionx*70,resolutiony*0.32])
151 controls=font.render("SPACE = hard drop the current figure",True,(255,125,0))
152 screen.blit(controls1,[resolutionx*70,resolutiony*0.35])
153 controls=font.render("ENTER = pause the game",True,(255,125,0))
154 screen.blit(controls1,[resolutionx*70,resolutiony*0.38])
155 controls=font.render("ESC = quit the game to desktop",True,(255,125,0))
156 screen.blit(controls1,[resolutionx*70,resolutiony*0.41])
157 controls=font.render("R = restart the game",True,(255,125,0))
158 screen.blit(controls1,[resolutionx*70,resolutiony*0.44])
159 text = font.render("Score: " + str(game.score), True, WHITE)
160 text_game_over = font.render("Game Over", True, (255, 125, 0))
161 text_game_over1 = font.render("Press SPACE to start again", True, (255, 125, 0))

```

Run

C:\Users\Razvan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Razvan\PycharmProjects\tetris-atestat\tetris.py
pygame 2.1.2 (SDL 2.0.18, Python 3.7.7)
Hello from the pygame community. <https://www.pygame.org/contribute.html>
True
Process finished with exit code 0


```

151 #text_game_over2 = font.render("Press ESCAPE to quit", True, (255,215,0))
152 text2 = font.render("Level: " + str(game.level), True, WHITE)
153 screen.blit(text, [resolutionx/2.75, resolutiony/7])
154 screen.blit(text2, [resolutionx/2.75, resolutiony/7*1.2])
155 textfig=font.render("next piece", True, WHITE)
156 screen.blit(textfig, [resolutionx/2*1.15, resolutiony/7*1.2])
157 thisisscoreboard=BigScoreboardfont.render("Scoreboard", True, RED)
158 screen.blit(thisisscoreboard, [resolutionx/2*1.1, resolutiony/5*1.05])
159 pygame.draw.rect(screen, (0,0,255), [resolutionx/2*1.15, resolutiony/5*1.1, game.zoom*7, game.zoom*10], 2)
160 scoreboarditerator=0
161 for scoreboarditerator in range(10):
162     scoreboardPlayer=scoreboardfont.render(str(scoreboarditerator+1)+" "+str(scoreboard_data[scoreboarditerator].playername)+" : "+str(scoreboard_data[scoreboarditerator+1]*25)), True, WHITE)
163     screen.blit(scoreboardPlayer, [resolutionx/2*1.15, resolutiony/5*1.07+(scoreboarditerator+1)*25])
164     scoreboarditerator=scoreboarditerator+1
165 if game.state == "gameover":
166     screen.blit(text_game_over, [resolutionx/2.5, resolutiony/4])
167     #screen.blit(text_game_over1, [resolutionx/2.5, resolutiony/4*1.25])
168     #screen.blit(text_game_over2, [resolutionx/2.5, resolutiony/4*1.35])
169     game.level=1
170 if (game.state=="recordscore"):
171     pygame.draw.rect(screen, GRAY, input_rect)
172     newrecord=fontbegin.render("What is your name?", True, GOLD)
173     screen.blit(newrecord, [resolutionx/3*1.2, resolutiony/3*0.8])
174     write = startfont.render(playername, True, GOLD)
175     pygame.draw.rect(screen, BLACK, [resolutionx/3*1.2, resolutiony/3, 500, 60])
176     pygame.draw.rect(screen, GOLD, [resolutionx/3*1.2, resolutiony/3, 500, 60], 1)
177     screen.blit(write, [resolutionx/3*1.2, resolutiony/3])
178     #schinbare difcultate
179     if game.score>10000:
180         game.level=2
181     if game.score>50000:
182         game.level=3
183     if game.score>90000:
184         game.level=4
185     if game.score>150000:
186         game.level=5
187     if game.score>200000:
188         game.level=6
189     pygame.display.flip()
190     clock.tick(fps)
191     if (quitgame):
192         pygame.quit()
193         break
194     pygame.quit()

```

```

151 #text_game_over2 = font.render("Press ESCAPE to quit", True, (255,215,0))
152 text2 = font.render("Level: " + str(game.level), True, WHITE)
153 screen.blit(text, [resolutionx/2.75, resolutiony/7])
154 screen.blit(text2, [resolutionx/2.75, resolutiony/7*1.2])
155 textfig=font.render("next piece", True, WHITE)
156 screen.blit(textfig, [resolutionx/2*1.15, resolutiony/7*1.2])
157 thisisscoreboard=BigScoreboardfont.render("Scoreboard", True, RED)
158 screen.blit(thisisscoreboard, [resolutionx/2*1.1, resolutiony/5*1.05])
159 pygame.draw.rect(screen, (0,0,255), [resolutionx/2*1.15, resolutiony/5*1.1, game.zoom*7, game.zoom*10], 2)
160 scoreboarditerator=0
161 for scoreboarditerator in range(10):
162     scoreboardPlayer=scoreboardfont.render(str(scoreboarditerator+1)+" "+str(scoreboard_data[scoreboarditerator].playername)+" : "+str(scoreboard_data[scoreboarditerator+1]*25)), True, WHITE)
163     screen.blit(scoreboardPlayer, [resolutionx/2*1.15, resolutiony/5*1.07+(scoreboarditerator+1)*25])
164     scoreboarditerator=scoreboarditerator+1
165 if game.state == "gameover":
166     screen.blit(text_game_over, [resolutionx/2.5, resolutiony/4])
167     #screen.blit(text_game_over1, [resolutionx/2.5, resolutiony/4*1.25])
168     #screen.blit(text_game_over2, [resolutionx/2.5, resolutiony/4*1.35])
169     game.level=1
170 if (game.state=="recordscore"):
171     pygame.draw.rect(screen, GRAY, input_rect)
172     newrecord=fontbegin.render("What is your name?", True, GOLD)
173     screen.blit(newrecord, [resolutionx/3*1.2, resolutiony/3*0.8])
174     write = startfont.render(playername, True, GOLD)
175     pygame.draw.rect(screen, BLACK, [resolutionx/3*1.2, resolutiony/3, 500, 60])
176     pygame.draw.rect(screen, GOLD, [resolutionx/3*1.2, resolutiony/3, 500, 60], 1)
177     screen.blit(write, [resolutionx/3*1.2, resolutiony/3])
178     #schinbare difcultate
179     if game.score>10000:
180         game.level=2
181     if game.score>50000:
182         game.level=3
183     if game.score>90000:
184         game.level=4
185     if game.score>150000:
186         game.level=5
187     if game.score>200000:
188         game.level=6
189     pygame.display.flip()
190     clock.tick(fps)
191     if (quitgame):
192         pygame.quit()
193         break
194     pygame.quit()

```


Bibliografie

- <https://www.codecademy.com/catalog/language/python>
- <https://www.jetbrains.com/help/pycharm/quick-start-guide.html#ui>
- <https://www.w3schools.com/python/>