**import** os  
**import** pygame  
**from** random **import** choice **as** rand  
  
pygame.init()  
  
FPS = 16  
step\_p = 10  
step\_g = 5  
score = 0  
pound\_score = 15  
level\_n = 1  
Width = 770  
Height = 890  
d\_w = 40  
d\_h = 40  
  
screen = pygame.display.set\_mode((Width, Height))  
clock = pygame.time.Clock()  
  
  
**def** randomase(vectors):  
 vectors1 = []  
 **for** i **in** range(4):  
 vectors1.append(rand(vectors))  
 vectors = vectors1  
  
  
**def** cor\_pos(self, vector):  
 **if** vector == 0:  
 **if** level[self.y][self.x + 1] != **'/'**:  
 **return True  
 else**:  
 **return False  
 if** vector == 90:  
 **if** level[self.y - 1][self.x] != **'/'**:  
 **return True  
 else**:  
 **return False  
 if** vector == 180:  
 **if** level[self.y][self.x - 1] != **'/'**:  
 **return True  
 else**:  
 **return False  
 if** vector == 270:  
 **if** level[self.y + 1][self.x] != **'/'**:  
 **return True  
 else**:  
 **return False  
  
  
def** time():  
 **global** ticks, secund, minute  
 ticks += 1  
 **if** ticks == FPS:  
 secund += 1  
 ticks = 0  
 **if** secund == 60:  
 minute += 1  
 secund = 0  
  
  
**def** load\_image(name, colorkey=**None**):  
 fullname = os.path.join(**'data'**, name)  
 **try**:  
 image = pygame.image.load(fullname)  
 **except** pygame.error **as** message:  
 print(**'Cannot load image:'**, name)  
 **raise** SystemExit(message)  
 image = image.convert\_alpha()  
 **if** colorkey **is not None**:  
 **if** colorkey **is** -1:  
 colorkey = image.get\_at((0, 0))  
 image.set\_colorkey(colorkey)  
 **return** image  
  
  
**def** load\_level(filename):  
 filename = **"data/"** + filename  
 **with** open(filename, **'r'**) **as** mapFile:  
 level\_map = [line.strip() **for** line **in** mapFile]  
  
 max\_width = 17  
  
 **return** list(map(**lambda** x: x.ljust(max\_width, **' '**), level\_map))  
  
  
screen\_sprite = pygame.sprite.Group()  
sprite = pygame.sprite.Sprite()  
sprite.image = load\_image(**"pac-man\_screen\_n.png"**)  
sprite.rect = sprite.image.get\_rect()  
sprite.rect.topleft = (40, 120)  
screen\_sprite.add(sprite)  
wall\_group = pygame.sprite.Group()  
all\_sprites = pygame.sprite.Group()  
tiles\_group = pygame.sprite.Group()  
player\_group = pygame.sprite.Group()  
ghosts\_group = pygame.sprite.Group()  
  
  
**class** Tile(pygame.sprite.Sprite):  
 **def** \_\_init\_\_(self, tile\_type, x, y):  
 super().\_\_init\_\_(tiles\_group, all\_sprites)  
 self.type = tile\_type  
 **if** tile\_type == **'pound'**:  
 image = pygame.Surface((40, 40), pygame.SRCALPHA, 32)  
 r = 4  
 pygame.draw.circle(image, pygame.Color(**"yellow"**), (4, 4), 4)  
 self.rect = pygame.Rect(x, y, 10, 10)  
 self.rect.x = d\_w \* x + 20 - r  
 self.rect.y = d\_h \* y + 20 - r  
 **if** tile\_type == **'energizer'**:  
 image = pygame.Surface((40, 40), pygame.SRCALPHA, 32)  
 r = 9  
 pygame.draw.circle(image, pygame.Color(**"white"**), (9, 9), 9)  
 self.rect = pygame.Rect(x, y, 20, 20)  
 self.rect.x = d\_w \* x + 20 - r  
 self.rect.y = d\_h \* y + 20 - r  
 self.image = image  
  
  
**class** Blinky(pygame.sprite.Sprite):  
 **def** \_\_init\_\_(self, ticks, start\_pound=0):  
 self.image = load\_image(**'Blinky.png'**)  
 self.type = **'Blinky'** self.vector = 180  
 self.run = **True** self.vectors = list(range(2, -3, -1))  
 self.ticks = ticks  
 self.start\_pound = start\_pound  
 self.x = 9  
 self.y = 9  
 self.act\_time = 0  
 self.fl = **True** self.image\_initialization()  
  
 **def** image\_initialization(self):  
 super().\_\_init\_\_(ghosts\_group)  
 self.rect = pygame.Rect(9, 11, 38, 38)  
 self.rect.x = d\_w \* self.x + 1  
 self.rect.y = d\_h \* self.y + 1  
  
 **def** update(self):  
 **if** self.vector == 180:  
 self.rect.x -= step\_g  
 **if** self.vector == 0:  
 self.rect.x += step\_g  
 **if** self.vector == 90:  
 self.rect.y -= step\_g  
 **if** self.vector == 270:  
 self.rect.y += step\_g  
  
 **def** uppdate\_pos(self):  
 **if** self.vector == 0:  
 **if** level[self.y][self.rect.x // d\_w + 1] == **'P'**:  
 self.x = 1  
 **else**:  
 self.x = self.rect.x // d\_w  
 **if** level[self.y][self.rect.x // d\_w + 1] == **'/'**:  
 self.fl = **False  
 elif** self.vector == 90:  
 self.y = self.rect.y // d\_w  
 **if** level[self.rect.y // d\_w - 1][self.x] == **'/'**:  
 self.fl = **False  
 elif** self.vector == 180:  
 **if** level[self.y][self.rect.x // d\_w - 1] == **'P'**:  
 self.x = 17  
 **else**:  
 self.x = self.rect.x // d\_w  
 **if** level[self.y][self.rect.x // d\_w - 1] == **'/'**:  
 self.fl = **False  
 elif** self.vector == 270:  
 self.y = self.rect.y // d\_w  
 **if** level[self.rect.y // d\_w + 1][self.x] == **'/'**:  
 self.fl = **False** self.rect.topleft = (self.x \* d\_w + 1, self.y \* d\_h + 1)  
  
 **def** uppdate\_vector(self):  
 **if** self.start\_pound - pound < self.act\_time **and not** self.run:  
 **if not** self.fl:  
 self.vector = (180 + self.vector) % 360  
 self.fl = **True  
 if** self.start\_pound - pound >= self.act\_time **and not** self.run:  
 self.y = 9  
 self.rect.y = 9 \* d\_h + 1  
 self.run = **True  
 if** self.run:  
 **if** self.vector == 90 **or** self.vector == 270:  
 **if** self.fl:  
 vector = rand([0, 180, self.vector])  
 **while not** cor\_pos(self, vector):  
 vector = rand([0, self.vector, 180])  
 self.vector = vector  
 **else**:  
 vector = rand([0, 180])  
 **while not** cor\_pos(self, vector):  
 vector = rand([0, 180])  
 self.vector = vector  
 self.fl = **True  
 elif** self.vector == 0 **or** self.vector == 180:  
 **if** self.fl:  
 vector = rand([90, 270, self.vector])  
 **while not** cor\_pos(self, vector):  
 vector = rand([90, self.vector, 270])  
 self.vector = vector  
 **else**:  
 vector = rand([90, 270])  
 **while not** cor\_pos(self, vector):  
 vector = rand([90, 270])  
 self.vector = vector  
 self.fl = **True  
  
  
class** Pinky(Blinky):  
 **def** \_\_init\_\_(self, ticks, start\_pound=0):  
 self.image = load\_image(**'Pinky.png'**)  
 self.type = **'Pinky'** self.vector = 180  
 self.run = **False** self.vectors = list(range(2, -3, -1))  
 self.ticks = ticks  
 self.start\_pound = start\_pound  
 self.act\_time = 5  
 self.x = 8  
 self.y = 11  
 self.fl = **True** self.image\_initialization()  
  
  
**class** Inky(Blinky):  
 **def** \_\_init\_\_(self, ticks, start\_pound=0):  
 self.image = load\_image(**'Inky.png'**)  
 self.type = **'Inky'** self.vector = 180  
 self.run = **False** self.vectors = list(range(2, -3, -1))  
 self.ticks = ticks  
 self.start\_pound = start\_pound  
 self.act\_time = 10  
 self.x = 10  
 self.y = 11  
 self.fl = **True** self.image\_initialization()  
  
  
**class** Clyde(Blinky):  
 **def** \_\_init\_\_(self, ticks, start\_pound=0):  
 self.image = load\_image(**'Clyde.png'**)  
 self.type = **'Clyde'** self.vector = 0  
 self.run = **False** self.vectors = list(range(2, -3, -1))  
 self.ticks = ticks  
 self.start\_pound = start\_pound  
 self.act\_time = 15  
 self.x = 9  
 self.y = 11  
 self.fl = **True** self.image\_initialization()  
  
  
**class** Player(pygame.sprite.Sprite):  
 **def** \_\_init\_\_(self, x=9, y=17):  
 super().\_\_init\_\_(player\_group)  
 self.image = load\_image(**'pac-man1.png'**)  
 self.rect = pygame.Rect(x, y, 38, 38)  
 self.vector = 180  
 self.vector1 = self.vector  
 self.ticks = 0  
 self.k = 1  
 self.x = x  
 self.y = y  
 self.fl = **True** self.rect.x = d\_w \* x + 1  
 self.rect.y = d\_h \* y + 1  
  
 **def** update(self):  
 **if** self.vector == 180:  
 self.rect.x -= step\_p  
 **if** self.vector == 0:  
 self.rect.x += step\_p  
 **if** self.vector == 90:  
 self.rect.y -= step\_p  
 **if** self.vector == 270:  
 self.rect.y += step\_p  
  
 **def** uppdate\_pos(self):  
 **if** self.vector == 0:  
 **if** level[self.y][self.rect.x // d\_w + 1] == **'P'**:  
 self.x = 1  
 **else**:  
 self.x = self.rect.x // d\_w  
 **if** level[self.y][self.rect.x // d\_w + 1] == **'/'**:  
 self.fl = **False  
 elif** self.vector == 90:  
 self.y = self.rect.y // d\_w  
 **if** level[self.rect.y // d\_w - 1][self.x] == **'/'**:  
 self.fl = **False  
 elif** self.vector == 180:  
 **if** level[self.y][self.rect.x // d\_w - 1] == **'P'**:  
 self.x = 17  
 **else**:  
 self.x = self.rect.x // d\_w  
 **if** level[self.y][self.rect.x // d\_w - 1] == **'/'**:  
 self.fl = **False  
 elif** self.vector == 270:  
 self.y = self.rect.y // d\_w  
 **if** level[self.rect.y // d\_w + 1][self.x] == **'/'**:  
 self.fl = **False** self.rect.topleft = (self.x \* d\_w + 1, self.y \* d\_h + 1)  
  
 **def** uppdate\_vector(self, vector1=**None**):  
 fl = **False  
 if** vector1 == **None**:  
 fl = **True** vector1 = self.vector1  
 self.vector1 = vector1  
 **if** self.vector1 != self.vector **and** (fl **or not** self.fl):  
 **if** self.vector1 == 0 **and** level[self.y][self.x + 1] != **'/'**:  
 self.fl = **True** self.ticks = ticks  
 self.vector = self.vector1  
 **elif** self.vector1 == 90 **and** level[self.y - 1][self.x] != **'/'**:  
 self.fl = **True** self.ticks = ticks  
 self.vector = self.vector1  
 **elif** self.vector1 == 180 **and** level[self.y][self.x - 1] != **'/'**:  
 self.fl = **True** self.ticks = ticks  
 self.vector = self.vector1  
 **elif** self.vector1 == 270 **and** level[self.y + 1][self.x] != **'/'**:  
 self.fl = **True** self.ticks = ticks  
 self.vector = self.vector1  
  
  
**def** generate\_level(level):  
 new\_player, x, y = **None**, **None**, **None  
 for** y **in** range(len(level)):  
 **for** x **in** range(len(level[y])):  
 **if** level[y][x] == **'.'**:  
 Tile(**'pound'**, x, y)  
 **if** level[y][x] == **'e'**:  
 Tile(**'energizer'**, x, y)  
 **elif** level[y][x] == **'@'**:  
 new\_player = Player(x, y)  
 **return** new\_player, x, y  
  
  
lives = [**'pac-man2.png'**, **'pac-man2.png'**, **'pac-man2.png'**]  
mode = [**'stabil'**, **'go'**]  
ticks = 0  
secund = 0  
minute = 0  
wait = 1 \* FPS  
level = load\_level(**'test\_level.txt'**)  
running = **True**end\_game = **False**pause = **False**running\_bonus = **False**pygame.display.flip()  
deith = **False  
while** running:  
 dead\_ghost = []  
 **if** end\_game:  
 **if** wait != 0:  
 wait -= 1  
 **continue  
 else**:  
 **break** pound = 152  
 pygame.display.flip()  
 player, width, height = generate\_level(level)  
 blinky = Blinky(ticks)  
 pinky = Pinky(ticks, pound)  
 inky = Inky(ticks, pound)  
 clyde = Clyde(ticks, pound)  
 screen\_sprite.draw(screen)  
 all\_sprites.draw(screen)  
 player\_group.draw(screen)  
 ghosts\_group.draw(screen)  
 f2 = pygame.font.SysFont(**'serif'**, 75)  
 text2 = f2.render(**"LEVEL "** + str(level\_n), 0, (255, 255, 0))  
 screen.blit(text2, (250, 30))  
 pygame.display.flip()  
 **if** pause:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.KEYDOWN **and** event.key == pygame.K\_SPACE:  
 pause = **False  
 continue  
 if** wait != 0:  
 clock.tick(FPS)  
 wait -= 1  
 all\_sprites = pygame.sprite.Group()  
 tiles\_group = pygame.sprite.Group()  
 player\_group = pygame.sprite.Group()  
 ghosts\_group = pygame.sprite.Group()  
 **continue** running\_level = **True** kill\_event = **False  
 if** level\_n == 4:  
 f3 = pygame.font.SysFont(**'serif'**, 80)  
 text3 = f3.render(**"You win!!!"**, 0, (255, 0, 0))  
 screen.blit(text3, (200, 450))  
 f4 = pygame.font.SysFont(**'serif'**, 50)  
 text4 = f4.render(**"Your score: "** + str(score), 0, (255, 255, 0))  
 screen.blit(text4, (250, 540))  
 pygame.display.flip()  
 running\_level = **False** end\_game = **True** running\_bonus = **True  
 if** level\_n == 2:  
 step\_g = 10  
 **if** level\_n == 3:  
 step\_p = 5  
 **while** running\_level:  
 **if** pause:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running = **False** running\_level = **False  
 if** event.type == pygame.KEYDOWN **and** event.key == pygame.K\_SPACE:  
 pause = **False  
 continue  
 if** wait != 0:  
 clock.tick(FPS)  
 wait -= 1  
 **continue  
 if** len(lives) > 1 **and** kill\_event:  
 kill\_event = **False** player\_group = pygame.sprite.Group()  
 ghosts\_group = pygame.sprite.Group()  
 dead\_ghost = []  
 blinky = Blinky(ticks, pound)  
 pinky = Pinky(ticks, pound)  
 inky = Inky(ticks, pound)  
 clyde = Clyde(ticks, pound)  
 player = Player()  
 **del** lives[-1]  
 wait = 2 \* FPS  
 **continue  
 elif** len(lives) == 1 **and** kill\_event:  
 f3 = pygame.font.SysFont(**'serif'**, 80)  
 text3 = f3.render(**"Game over"**, 0, (255, 0, 0))  
 screen.blit(text3, (200, 450))  
 f4 = pygame.font.SysFont(**'serif'**, 50)  
 text4 = f4.render(**"Your score: "** + str(score), 0, (255, 255, 0))  
 screen.blit(text4, (250, 540))  
 pygame.display.flip()  
 wait = 3 \* FPS  
 dead\_ghost = []  
 end\_game = **True  
 break  
 if** len(tiles\_group) == 0:  
 wait = 2 \* FPS  
 player.k = 1  
 level\_n += 1  
 dead\_ghost = []  
 mode = [**'stabil'**, **'go'**]  
 **break  
 for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running = **False** running\_level = **False  
 if** event.type == pygame.KEYDOWN **and** event.key == pygame.K\_SPACE:  
 pause = **True  
 if** event.type == pygame.KEYDOWN:  
 **if** event.key == pygame.K\_LEFT:  
 player.uppdate\_vector(180)  
 **if** event.key == pygame.K\_RIGHT:  
 player.uppdate\_vector(0)  
 **if** event.key == pygame.K\_UP:  
 player.uppdate\_vector(90)  
 **if** event.key == pygame.K\_DOWN:  
 player.uppdate\_vector(270)  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_p) == player.ticks % (  
 40 // step\_p):  
 player.uppdate\_pos()  
 player.uppdate\_vector()  
 player.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == blinky.ticks % (  
 40 // step\_g):  
 blinky.uppdate\_pos()  
 blinky.uppdate\_vector()  
 blinky.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == pinky.ticks % (  
 40 // step\_g):  
 pinky.uppdate\_pos()  
 pinky.uppdate\_vector()  
 pinky.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == inky.ticks % (  
 40 // step\_g):  
 inky.uppdate\_pos()  
 inky.uppdate\_vector()  
 inky.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == clyde.ticks % (  
 40 // step\_g):  
 clyde.uppdate\_pos()  
 clyde.uppdate\_vector()  
 clyde.uppdate\_pos()  
 **if** ticks % 4 <= 1:  
 player.image = load\_image(**'pac-man1.png'**)  
 **elif** ticks % 4 > 1:  
 player.image = load\_image(**'pac-man2.png'**)  
 **if** player.vector != 180:  
 player.image = pygame.transform.rotate(player.image, player.vector)  
 **else**:  
 player.image = pygame.transform.flip(player.image, **True**, **False**)  
 screen.fill((0, 0, 0))  
 screen\_sprite.draw(screen)  
 all\_sprites.draw(screen)  
 **if** player.fl:  
 player.update()  
 **if** pygame.sprite.spritecollideany(player, all\_sprites):  
 food = pygame.sprite.spritecollide(player, tiles\_group, **True**)  
 **for** i **in** food:  
 **if** i.type == **'pound'**:  
 score += pound\_score  
 pound -= 1  
 **elif** i.type == **'energizer'**:  
 score += pound\_score \* 5  
 mode = [**'rush'**, **'scare'**, 10 \* FPS]  
 player.k = 1  
 **if** mode[0] == **'rush' and** pygame.sprite.spritecollideany(player,  
 ghosts\_group):  
 dying\_goost = pygame.sprite.spritecollide(player, ghosts\_group, **True**)  
 **for** i **in** dying\_goost:  
 score += (2 \*\* player.k) \* 100  
 player.k += 1  
 dead\_ghost.append([i.type, 15 \* FPS])  
 **del** i  
 **if** mode[0] == **'rush' and** mode[2] == 0:  
 mode = [**'stabil'**, **'go'**]  
 player.k = 1  
 **if** mode[1] == **'scare'**:  
 mode[2] -= 1  
 f0 = pygame.font.SysFont(**None**, 20)  
 text0 = f0.render(str(mode[2] // FPS), 0, (255, 0, 0))  
 screen.blit(text0, (740, 50))  
 blinky.image = load\_image(**'rush1.png'**)  
 pinky.image = load\_image(**'rush1.png'**)  
 inky.image = load\_image(**'rush1.png'**)  
 clyde.image = load\_image(**'rush1.png'**)  
 **if** mode[0] == **'rush'**:  
 step\_p = 10  
 **if** mode[0] == **'stabil' and** level\_n == 3:  
 step\_p = 5  
 **if** mode[1] == **'go'**:  
 blinky.image = load\_image(**'Blinky.png'**)  
 pinky.image = load\_image(**'Pinky.png'**)  
 inky.image = load\_image(**'Inky.png'**)  
 clyde.image = load\_image(**'Clyde.png'**)  
 **for** s **in** range(len(dead\_ghost)):  
 i = dead\_ghost[s]  
 **if** i[1] != 0:  
 i[1] -= 1  
 **elif** i[1] == 0:  
 i[1] -= 1  
 **if** i[0] == **'Blinky'**:  
 blinky = Blinky(ticks, pound)  
 **if** i[0] == **'Pinky'**:  
 pinky = Pinky(ticks, pound)  
 pinky.rect.topleft = (9 \* d\_w, 11 \* d\_h)  
 **if** i[0] == **'Inky'**:  
 inky = Inky(ticks, pound)  
 **if** i[0] == **'Clyde'**:  
 clyde = Clyde(ticks, pound)  
 dead\_ghost = list(filter(**lambda** x: x[1] >= 0, dead\_ghost))  
 **if** blinky.fl:  
 blinky.update()  
 **if** pinky.fl:  
 pinky.update()  
 **if** inky.fl:  
 inky.update()  
 **if** clyde.fl:  
 clyde.update()  
 **if** mode[0] == **'stabil' and** pygame.sprite.spritecollideany(player, ghosts\_group):  
 kill\_event = **True** mode = [**'stabil'**, **'go'**]  
 player\_group.draw(screen)  
 ghosts\_group.draw(screen)  
  
 f1 = pygame.font.SysFont(**'serif'**, 30)  
 text1 = f1.render(**"Score: "** + str(score), 0, (255, 255, 0))  
 screen.blit(text1, (50, 20))  
  
 f2 = pygame.font.SysFont(**'serif'**, 75)  
 text2 = f2.render(**"LEVEL "** + str(level\_n), 0, (255, 255, 0))  
 screen.blit(text2, (250, 30))  
  
 lives\_group = pygame.sprite.Group()  
 **for** x **in** range(len(lives)):  
 sprite = pygame.sprite.Sprite()  
 sprite.image = load\_image(lives[x])  
 sprite.rect = sprite.image.get\_rect()  
 sprite.rect.topleft = (600 + x \* 40, 20)  
 lives\_group.add(sprite)  
  
 sprite = pygame.sprite.Sprite()  
 sprite.image = pygame.Surface((40, 5), pygame.SRCALPHA, 32)  
 pygame.draw.rect(sprite.image, pygame.Color(**"blue"**), (0, 0, 40, 5))  
 sprite.rect = sprite.image.get\_rect()  
 sprite.rect.topleft = (600 + x \* 40, 60)  
 lives\_group.add(sprite)  
  
 lives\_group.draw(screen)  
  
 time()  
 clock.tick(FPS)  
 pygame.display.flip()  
  
 all\_sprites = pygame.sprite.Group()  
 tiles\_group = pygame.sprite.Group()  
 player\_group = pygame.sprite.Group()  
 ghosts\_group = pygame.sprite.Group()  
 screen.fill((0, 0, 0))  
 clock.tick(FPS)  
end\_game = **False**level\_n = 4  
energ = 0  
win = **False**wait = 3 \* FPS  
**while** running\_bonus:  
 **if** end\_game:  
 **if** wait != 0:  
 wait -= 1  
 **else**:  
 **break** pound = 152  
 pygame.display.flip()  
 player, width, height = generate\_level(level)  
 blinky = Blinky(ticks, pound)  
 pinky = Pinky(ticks, pound)  
 inky = Inky(ticks, pound)  
 clyde = Clyde(ticks, pound)  
 screen\_sprite.draw(screen)  
 all\_sprites.draw(screen)  
 player\_group.draw(screen)  
 ghosts\_group.draw(screen)  
 f2 = pygame.font.SysFont(**'serif'**, 60)  
 text2 = f2.render(**"BONUS LEVEL"**, 0, (255, 0, 0))  
 screen.blit(text2, (185, 50))  
 pygame.display.flip()  
 clock.tick(FPS)  
 **if** pause:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running\_bonus = **False** running\_level = **False  
 if** event.type == pygame.KEYDOWN **and** event.key == pygame.K\_SPACE:  
 pause = **False  
 continue** running\_level = **True** kill\_event = **False  
 if** wait != 0:  
 all\_sprites = pygame.sprite.Group()  
 tiles\_group = pygame.sprite.Group()  
 player\_group = pygame.sprite.Group()  
 ghosts\_group = pygame.sprite.Group()  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running\_bonus = **False** running\_level = **False** clock.tick(FPS)  
 wait -= 1  
 **continue  
 if** win:  
 f3 = pygame.font.SysFont(**'serif'**, 80)  
 text3 = f3.render(**"You're the best!!!"**, 0, (255, 0, 0))  
 screen.blit(text3, (120, 450))  
 f4 = pygame.font.SysFont(**'serif'**, 50)  
 text4 = f4.render(**"Your score: "** + str(score), 0, (255, 255, 0))  
 screen.blit(text4, (250, 540))  
 pygame.display.flip()  
 wait = 2 \* FPS  
 end\_game = **True  
 continue** step\_p = 5  
 step\_g = 10  
 **while** running\_level:  
 **if** pause:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running\_bonus = **False** running\_level = **False  
 if** event.type == pygame.KEYDOWN **and** event.key == pygame.K\_SPACE:  
 pause = **False  
 continue  
 if** wait != 0:  
 clock.tick(FPS)  
 wait -= 1  
 **continue  
 if** len(lives) > 1 **and** kill\_event:  
 kill\_event = **False** player\_group = pygame.sprite.Group()  
 ghosts\_group = pygame.sprite.Group()  
 blinky = Blinky(ticks)  
 pinky = Pinky(ticks, pound)  
 inky = Inky(ticks, pound)  
 clyde = Clyde(ticks, pound)  
 player = Player()  
 **del** lives[-1]  
 wait = 2 \* FPS  
 **continue  
 elif** len(lives) == 1 **and** kill\_event:  
 f3 = pygame.font.SysFont(**'serif'**, 80)  
 text3 = f3.render(**"Game over"**, 0, (255, 0, 0))  
 screen.blit(text3, (200, 450))  
 f4 = pygame.font.SysFont(**'serif'**, 50)  
 text4 = f4.render(**"Your score: "** + str(score), 0, (255, 255, 0))  
 screen.blit(text4, (250, 540))  
 pygame.display.flip()  
 wait = 2 \* FPS  
 end\_game = **True  
 break  
 if** len(tiles\_group) == 0:  
 player.k = 1  
 mode = [**'stabil'**, **'go'**]  
 win = **True  
 break  
 if** energ == 4:  
 step\_p = 10  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 running\_bonus = **False** running\_level = **False  
 if** event.type == pygame.KEYDOWN **and** event.key == pygame.K\_SPACE:  
 pause = **True  
 if** event.type == pygame.KEYDOWN:  
 **if** event.key == pygame.K\_LEFT:  
 player.uppdate\_vector(180)  
 **if** event.key == pygame.K\_RIGHT:  
 player.uppdate\_vector(0)  
 **if** event.key == pygame.K\_UP:  
 player.uppdate\_vector(90)  
 **if** event.key == pygame.K\_DOWN:  
 player.uppdate\_vector(270)  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_p) == player.ticks % (  
 40 // step\_p):  
 player.uppdate\_pos()  
 player.uppdate\_vector()  
 player.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == blinky.ticks % (  
 40 // step\_g):  
 blinky.uppdate\_pos()  
 blinky.uppdate\_vector()  
 blinky.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == pinky.ticks % (  
 40 // step\_g):  
 pinky.uppdate\_pos()  
 pinky.uppdate\_vector()  
 pinky.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == inky.ticks % (  
 40 // step\_g):  
 inky.uppdate\_pos()  
 inky.uppdate\_vector()  
 inky.uppdate\_pos()  
 **if** (ticks + secund \* FPS + minute \* 60 \* FPS) % (40 // step\_g) == clyde.ticks % (  
 40 // step\_g):  
 clyde.uppdate\_pos()  
 clyde.uppdate\_vector()  
 clyde.uppdate\_pos()  
 **if** ticks % 4 <= 1:  
 player.image = load\_image(**'pac-man1.png'**)  
 **elif** ticks % 4 > 1:  
 player.image = load\_image(**'pac-man2.png'**)  
 **if** player.vector != 180:  
 player.image = pygame.transform.rotate(player.image, player.vector)  
 **else**:  
 player.image = pygame.transform.flip(player.image, **True**, **False**)  
 screen.fill((0, 0, 0))  
 screen\_sprite.draw(screen)  
 all\_sprites.draw(screen)  
 **if** player.fl:  
 player.update()  
 **if** pygame.sprite.spritecollideany(player, all\_sprites):  
 food = pygame.sprite.spritecollide(player, tiles\_group, **True**)  
 **for** i **in** food:  
 **if** i.type == **'pound'**:  
 score += pound\_score  
 **elif** i.type == **'energizer'**:  
 **if** player.vector == 0:  
 player.x = 4  
 player.y = 15  
 player.vector = 180  
 player.vector1 = 180  
 **elif** player.vector == 180:  
 player.x = 14  
 player.y = 15  
 player.vector = 0  
 player.vector1 = 0  
 **elif** player.vector == 90:  
 player.x = 9  
 player.y = 21  
 player.vector = 180  
 player.vector1 = 180  
 **elif** player.vector == 270:  
 player.x = 9  
 player.y = 5  
 player.vector = 0  
 player.vector1 = 0  
 energ += 1  
 player.rect.topleft = (player.x \* d\_w, player.y \* d\_h)  
 score += pound\_score \* 5  
 **if** blinky.fl:  
 blinky.update()  
 **if** pinky.fl:  
 pinky.update()  
 **if** inky.fl:  
 inky.update()  
 **if** clyde.fl:  
 clyde.update()  
 **if** pygame.sprite.spritecollideany(player, ghosts\_group):  
 kill\_event = **True** player\_group.draw(screen)  
 ghosts\_group.draw(screen)  
  
 f1 = pygame.font.SysFont(**'serif'**, 30)  
 text1 = f1.render(**"Score: "** + str(score), 0, (255, 255, 0))  
 screen.blit(text1, (50, 20))  
  
 f2 = pygame.font.SysFont(**'serif'**, 60)  
 text2 = f2.render(**"BONUS LEVEL"**, 0, (255, 0, 0))  
 screen.blit(text2, (185, 50))  
  
 lives\_group = pygame.sprite.Group()  
 **for** x **in** range(len(lives)):  
 sprite = pygame.sprite.Sprite()  
 sprite.image = load\_image(lives[x])  
 sprite.rect = sprite.image.get\_rect()  
 sprite.rect.topleft = (600 + x \* 40, 20)  
 lives\_group.add(sprite)  
  
 sprite = pygame.sprite.Sprite()  
 sprite.image = pygame.Surface((40, 5), pygame.SRCALPHA, 32)  
 pygame.draw.rect(sprite.image, pygame.Color(**"blue"**), (0, 0, 40, 5))  
 sprite.rect = sprite.image.get\_rect()  
 sprite.rect.topleft = (600 + x \* 40, 60)  
 lives\_group.add(sprite)  
  
 lives\_group.draw(screen)  
  
 time()  
 clock.tick(FPS)  
 pygame.display.flip()  
  
 all\_sprites = pygame.sprite.Group()  
 tiles\_group = pygame.sprite.Group()  
 player\_group = pygame.sprite.Group()  
 ghosts\_group = pygame.sprite.Group()  
 clock.tick(FPS)  
**try**:  
 f = open(**'nicknames.txt'**).read()  
 s = int(f[len(f) - 2])  
**except** Exception:  
 open(**'nicknames.txt'**, **'a'**).write(**': '** + str(score) + **'\n'**)  
**else**:  
 file = open(**'nicknames.txt'**).read().split(**' '**)  
 file[-1] = str(score) + **'\n'** open(**'nicknames.txt'**, **'w'**).write(**' '**.join(file))  
pygame.quit()  
os.system(**'python {}'**.format(**'Title.py'**))