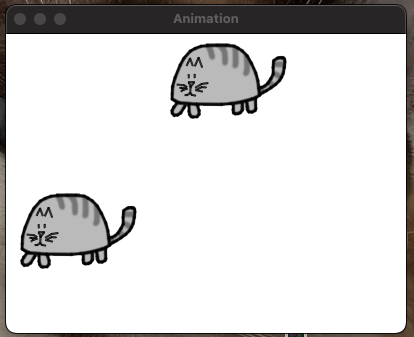
**(wiekszosc zadan robilam dawno temu i moge nie pamietac dokladnie wszystkich dodanych linijek.....)**

**Taks list:**

* **catanimation.py:**

Extend the game by adding second cat with different directions. Each cat should move independently and bounce off the edges of the screen



catImg2 = catImg.copy()

catx2 = 10

caty2 = 10

direction2 = 'down'

*if* direction2 == 'down':

caty2 += 5

*if* caty2 == 220:

direction2 = 'right'

*elif* direction2 == 'right':

catx2 += 5

*if* catx2 == 280:

direction2 = 'up'

*elif* direction2 == 'up':

caty2 -= 5

*if* caty2 == 10:

direction2 = 'left'

*elif* direction2 == 'left':

catx2 -= 5

*if* catx2 == 10:

direction2 = 'down'

DISPLAYSURF.blit(catImg2, (catx2, caty2))

* **simulate.py:**

Add a feature to display a "Game Over" message when the player loses.



font = pygame.font.Font('freesansbold.ttf',40)

surface = font.render('Game Over', True, WHITE)

rectangle = surface.get\_rect()

rectangle.center = (WINDOWWIDTH // 2, WINDOWHEIGHT // 2)

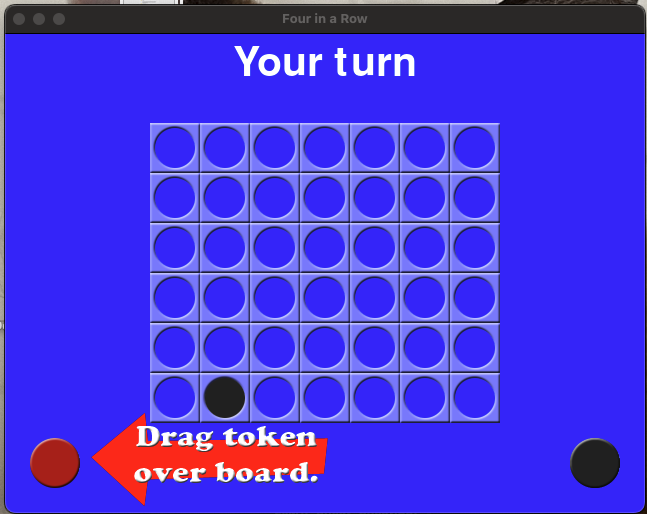
DISPLAYSURF.blit(surface, rectangle)

pygame.display.update()

pygame.time.wait(500)

* **fourinarow.py:**

Add a feature to display the current player's turn on the screen.



*if* *turn* == HUMAN:

font = pygame.font.Font(None, 60)

text = font.render('Your turn', True, WHITE)

text\_rect = text.get\_rect()

text\_rect.center = (WINDOWWIDTH // 2, 30)

DISPLAYSURF.blit(text, text\_rect)

* **squirrel.py:**

Add a feature to display a "Level Up" message and increase the game speed slightly every time the player reaches a score that is a multiple of 5



*if* sqObj['width'] \* sqObj['height'] <= playerObj['size']\*\*2:

oldScore = score

score += 1

*if* score % 5 == 0 and score > lastSpeedIncrease:

speedMultiplier += 0.2

lastSpeedIncrease = score

showLevelUp = True

levelUpStartTime = time.time()

levelUpSurf = BASICFONT.render('SPEED UP!', True, RED)

*elif* score > oldScore:

showLevelUp = True

levelUpStartTime = time.time()

levelUpSurf = BASICFONT.render('LEVEL UP!', True, RED)

scoreSurf = BASICFONT.render('Score: ' + str(score), True, RED)

scoreRect = scoreSurf.get\_rect()

scoreRect.topleft = (50, 15)

DISPLAYSURF.blit(scoreSurf, scoreRect)

*if* showLevelUp:

DISPLAYSURF.blit(levelUpSurf, levelUpRect)

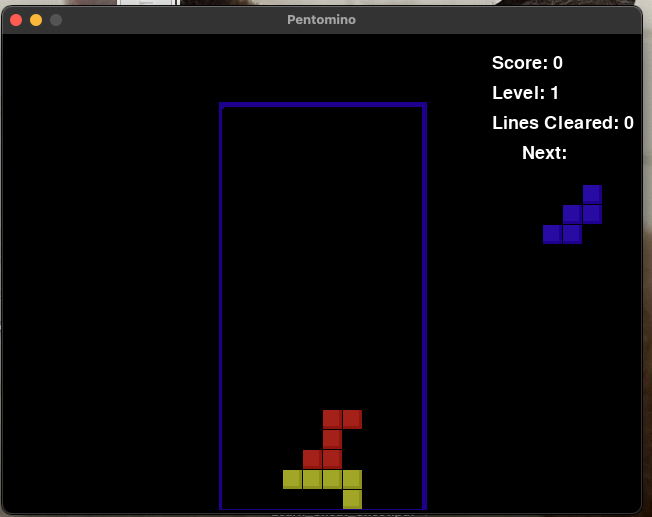
*if* time.time() - levelUpStartTime > 1.0:

showLevelUp = False

* **pentomino.py:**

Add a feature to display the number of lines cleared during the game in the top-right corner of the screen.

(nie umiem w to grac ale obiecuje ze liczba linii sie zwieksza)



lines\_cleared = 0

addToBoard(board, fallingPiece)

lines\_removed = removeCompleteLines(board)

lines\_cleared += lines\_removed

score += lines\_removed

lines\_cleared\_surf = BASICFONT.render('Lines Cleared: %s' % *lines\_cleared*, True, TEXTCOLOR)

lines\_cleared\_rect = lines\_cleared\_surf.get\_rect()

lines\_cleared\_rect.topleft = (WINDOWWIDTH - 150, 80)

DISPLAYSURF.blit(lines\_cleared\_surf, lines\_cleared\_rect)

* **gemgem.py:**

Add a feature to display a timer at the top of the screen that counts down from a set time limit (e.g., 2 minutes). When the timer reaches zero, the game ends.



GAME\_TIME\_LIMIT = 120

TIMER\_COLOR = RED

score = 0

start\_time = time.time()

time\_remaining = GAME\_TIME\_LIMIT

time\_remaining = max(0, GAME\_TIME\_LIMIT - int(time.time() - start\_time))

*if* time\_remaining == 0 and not gameIsOver:

gameIsOver = True

minutes = time\_remaining // 60

seconds = time\_remaining % 60

timerSurf = BASICFONT.render('Time: %02d:%02d' % (minutes, seconds), 1, TIMER\_COLOR)

timerRect = timerSurf.get\_rect()

timerRect.topleft = (10, 10)

DISPLAYSURF.blit(timerSurf, timerRect)

*if* time\_remaining == 0:

message = 'Time\'s Up! Final Score: %s (Click to continue)' % (score)