Wormy Game

https://github.com/zhihongzeng2002/pythongame

Game Begin

In the middle of game

Rules:

1. Eat the apple
2. Don’t hit the wall
3. Don’t hit worm body

Game over

Game Time: 10 minutes
https://github.com/zhihongzeng2002/pythongame

Winner: the highest score

Project 1: Training mode

Code: setup

```
import random, pygame, sys
from pygame.locals import *

FPS = 5
WINDOWWIDTH = 640
WINDOWHEIGHT = 480
CELLSIZE = 20
CELLWIDTH = int(WINDOWWIDTH / CELLSIZE)
CELLHEIGHT = int(WINDOWHEIGHT / CELLSIZE)

#               R    G    B
BGCOLOR     = (  0,   0,   0)
GREEN       = (  0, 255,   0)
DARKGREEN   = (  0, 155,   0)
DARKGRAY    = ( 40,  40,  40)

UP = 'up'
DOWN = 'down'
LEFT = 'left'
RIGHT = 'right'

HEAD = 0 # index of the worm's head
```

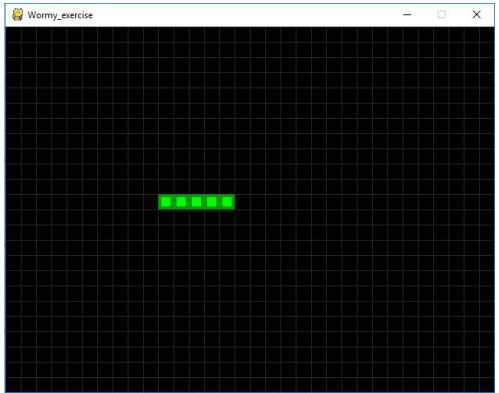
Main function

```
def main():
    global FPSCLOCK, DISPLAYSURF, BASICFONT

    pygame.init()
    FPSCLOCK = pygame.time.Clock()
    DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
    pygame.display.set_caption('Wormy_exercise')

    runGame()
```

Data structure of worm

```
def runGame():
    startx = random.randint(7, CELLWIDTH - 6)
    starty = random.randint(7, CELLHEIGHT - 6)
    wormCoords = [[startx, starty], [startx-1, starty],
        [startx-2, starty], [startx-3, starty], [startx-4, starty]]
    HEAD = 0
    direction = RIGHT
```

4	3	2	1	0



Main Loop: Event check

```
while True: # main game loop
    for event in pygame.event.get():
        if event.type == QUIT:
            terminate()
        elif event.type == KEYDOWN:
            if event.key == K_LEFT and direction != RIGHT:
                direction = LEFT
            elif event.key == K_RIGHT and direction != LEFT:
                direction = RIGHT
            elif event.key == K_UP and direction != DOWN:
                direction = UP
            elif event.key == K_DOWN and direction != UP:
                direction = DOWN
            else:
                break
```

Worm movement

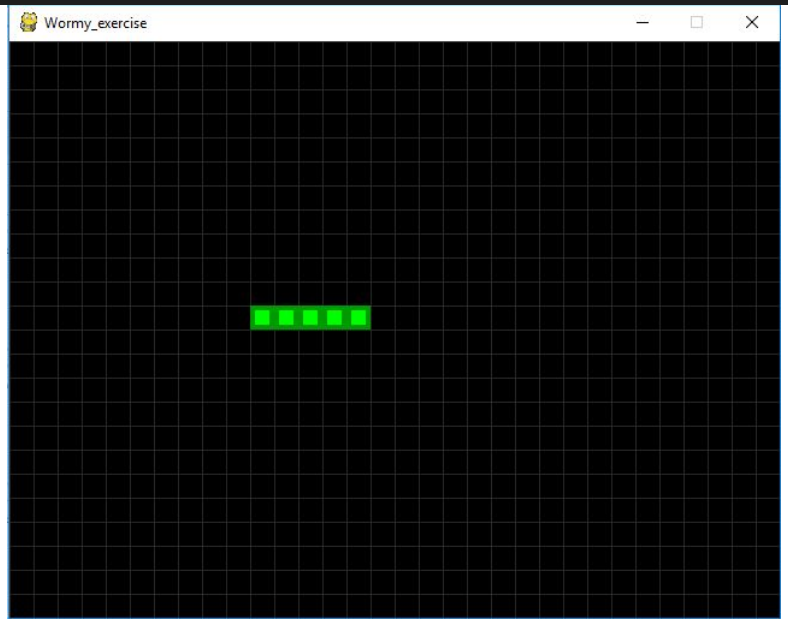
```
if direction == UP:
    newHead = [wormCoords[HEAD][0], wormCoords[HEAD][1]-1]
elif direction == DOWN:
    newHead = [wormCoords[HEAD][0], wormCoords[HEAD][1]+1]
elif direction == LEFT:
    newHead = [wormCoords[HEAD][0]-1, wormCoords[HEAD][1]]
elif direction == RIGHT:
    newHead = [wormCoords[HEAD][0]+1, wormCoords[HEAD][1]]

if newHead[0] >= 0 and newHead[0] < CELLWIDTH and \
        newHead[1] >= 0 and newHead[1] < CELLHEIGHT:
    wormCoords.insert(0, newHead)
    del wormCoords[-1]
```

4	3	2	1	0	new

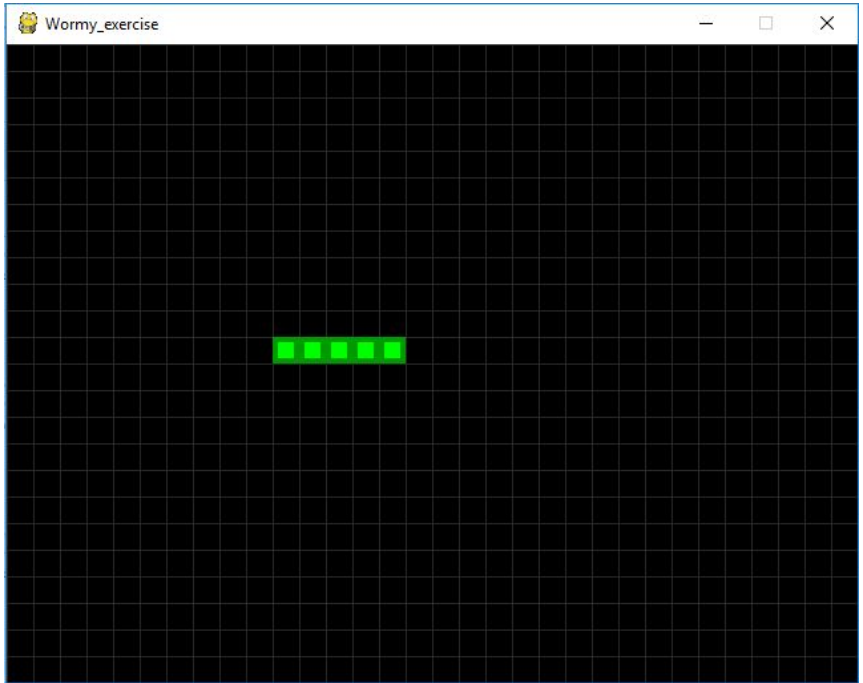
Main Loop: rendering

```
DISPLAYSURF.fill(BGCOLOR)
drawGrid()
drawWorm(wormCoords)
pygame.display.update()
FPSCLOCK.tick(FPS)
```

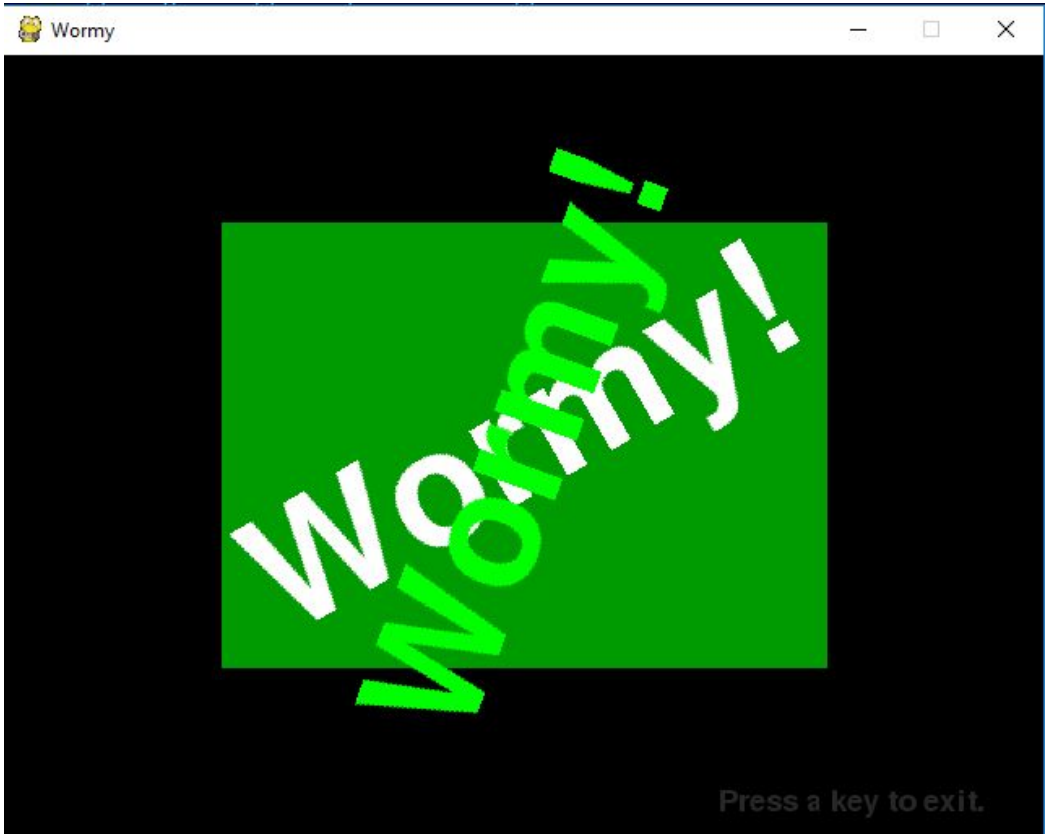
```
def drawWorm(wormCoords):
    for coord in wormCoords:
        x = coord[0] * CELLSIZE
        y = coord[1] * CELLSIZE
        wormSegmentRect = pygame.Rect(x, y, CELLSIZE, CELLSIZE)
        pygame.draw.rect(DISPLAYSURF, DARKGREEN, wormSegmentRect)
        wormInnerSegmentRect = pygame.Rect(x + 4, y + 4, CELLSIZE - 8, CELLSIZE - 8)
        pygame.draw.rect(DISPLAYSURF, GREEN, wormInnerSegmentRect)
```



```python
def drawGrid():
    for x in range(0, WINDOWWIDTH, CELLSIZE): # draw vertical lines
        pygame.draw.line(DISPLAYSURF, DARKGRAY, (x, 0), (x, WINDOWHEIGHT))
    for y in range(0, WINDOWHEIGHT, CELLSIZE): # draw horizontal lines
        pygame.draw.line(DISPLAYSURF, DARKGRAY, (0, y), (WINDOWWIDTH, y))
```



Project2: rotation



Begin:

```
import random, pygame, sys
from pygame.locals import *

FPS = 5
WINDOWWIDTH = 640
WINDOWHEIGHT = 480

def main():
    global FPSCLOCK, DISPLAYSURF

    pygame.init()
    FPSCLOCK = pygame.time.Clock()
    DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
    pygame.display.set_caption('Wormy')

    showStartScreen()
```

```
def showStartScreen():
    WHITE = (255, 255, 255)
    BGCOLOR     = (  0,   0,   0)
    GREEN     = (  0, 255,   0)
    DARKGREEN = (  0, 155,   0)

    titleFont = pygame.font.Font('freesansbold.ttf', 100)
    titleSurf1 = titleFont.render('Wormy!', True, WHITE, DARKGREEN)
    titleSurf2 = titleFont.render('Wormy!', True, GREEN)

    degrees1 = 0
    degrees2 = 0
```

```
while True:
    if checkForKeyPress():
        break

    DISPLAYSURF.fill(BGCOLOR)
    rotatedSurf1 = pygame.transform.rotate(titleSurf1, degrees1)
    rotatedRect1 = rotatedSurf1.get_rect()
    rotatedRect1.center = (WINDOWWIDTH / 2, WINDOWHEIGHT / 2)
    DISPLAYSURF.blit(rotatedSurf1, rotatedRect1)

    rotatedSurf2 = pygame.transform.rotate(titleSurf2, degrees2)
    rotatedRect2 = rotatedSurf2.get_rect()
    rotatedRect2.center = (WINDOWWIDTH / 2, WINDOWHEIGHT / 2)
    DISPLAYSURF.blit(rotatedSurf2, rotatedRect2)

    drawPressKeyMsg()

    pygame.display.update()
    FPSCLOCK.tick(FPS)
    degrees1 += 3 # rotate by 3 degrees each frame
    degrees2 += 7 # rotate by 7 degrees each frame
```

```
def drawPressKeyMsg():
    BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
    DARKGRAY  = ( 40,  40,  40)
    pressKeySurf = BASICFONT.render('Press a key to exit.', True, DARKGRAY)
    pressKeyRect = pressKeySurf.get_rect()
    pressKeyRect.topleft = (WINDOWWIDTH - 200, WINDOWHEIGHT - 30)
    DISPLAYSURF.blit(pressKeySurf, pressKeyRect)

def checkForKeyPress():
    if len(pygame.event.get(QUIT)) > 0:
        return True

    keyUpEvents = pygame.event.get(KEYUP)
    if len(keyUpEvents) == 0:
        return False
    else:
        return True

if __name__ == '__main__':
    main()
```