ספר הפרויקט

שם: אור אשרוב

ת"ז: [redacted]

בית ספר: מקיף עירוני ח'

מגמה: סייבר

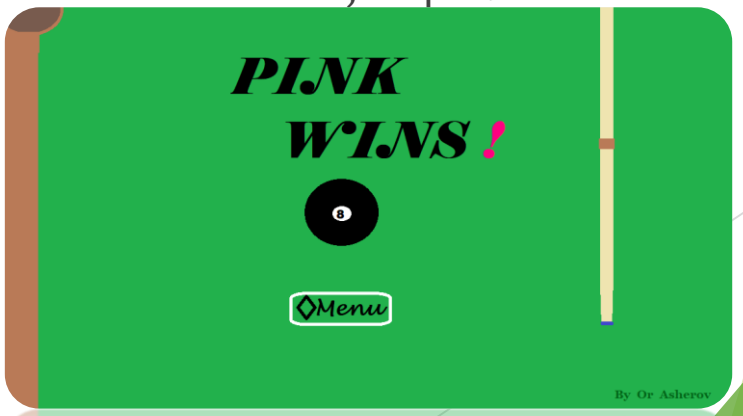
[redacted]

תוכן עניינים

מבוא

- עבודת הגמר שלי נכתבה בשפת המחשב Python.
- נעזרתי במודלים בספריה של פייתון : pygame, math.
- שם המשחק הוא "Billiard". זהו משחק ביליארד המיועד לשני שחקנים המתחרים אחד נגד השני, על מחשב אחד.
- השחקנים מתחלקים לשני צבעים: כחול וורוד וכך גם הכדורים (מלבד הכדור הלבן והכדור השחור).
- המשחק מסתיים כאשר אחד מהשחקנים מנצח (השחקן שצבעו הוא כחול או השחקן שצבעו ורוד) וניתן להתחילו מחדש על ידי לחיצה על מקש "Menu" במסך המודיע על ניצחון אחד השחקנים;

או

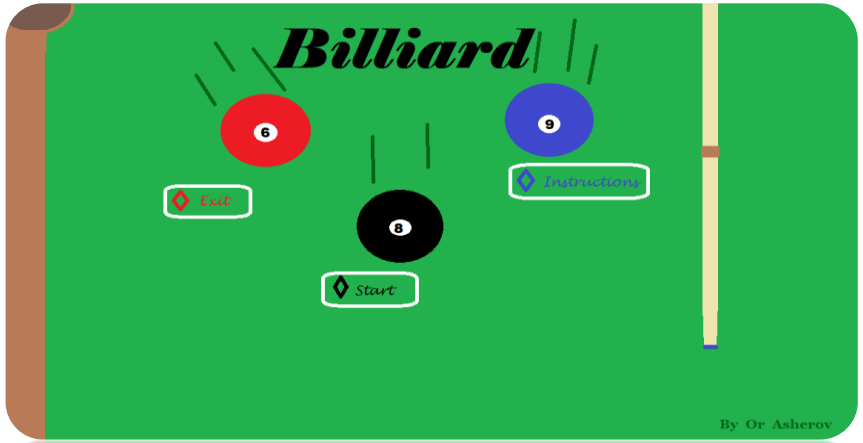


הפעולה שמציגה את מסך הניצחון:

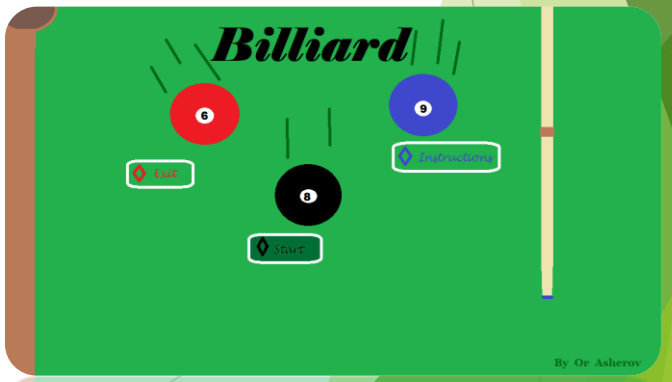
```
def draw_win_screen(win_screen, winner):
    music=pygame.mixer.Sound("winning.wav")
    image_selected = "Images/blue_selected.png"
    image = "Images/blue.png"
    if winner == "Blue":
        image_selected = "Images/blue_selected.png"
        image = "Images/blue.png"
    elif winner == "Pink":
        image_selected = "Images/pink_selected.png"
        image = "Images/pink.png"

    # get mouse position and click status
    mouse_x, mouse_y = pygame.mouse.get_pos()
    mouse_left = get_mouse_press()

    # if mouse is hovering over a button, then highlight that button
    if mouse_x > 470 and mouse_x < 550 and mouse_y > 400 and mouse_y < 600:
        myimage = pygame.image.load(image_selected)
        if mouse_left == True:
            pygame.mixer.Sound.stop(music)
            win_screen = False # go to main menu

    else:
        myimage = pygame.image.load(image)

    imagerect = myimage.get_rect()
    screen.fill(BLACK)
    screen.blit(myimage, imagerect)
    pygame.mixer.Sound.play(music)
    pygame.display.flip()

    return win_screen
```

מסך הפתיחה

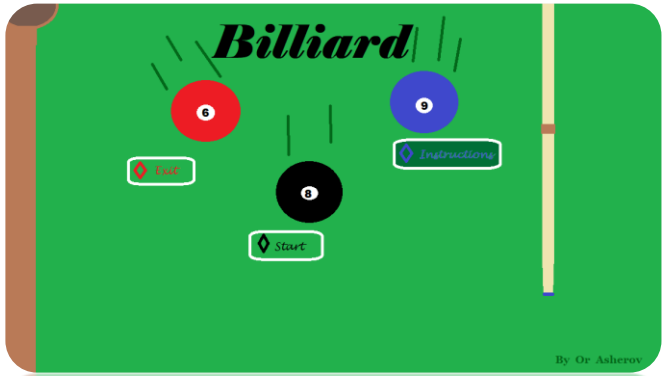
- מסך הפתיחה של המשחק:



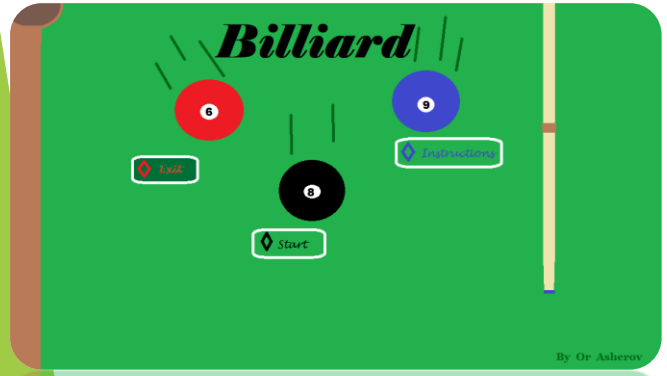
- כאשר העכבר נמצא על אחד הלחצנים (Start/Instructions/Exit) הלחצן יהפוך לכהה יותר (קובץ png שונה) וישמע צליל , כל עוד העכבר עדיין על הלחצן.



או



או

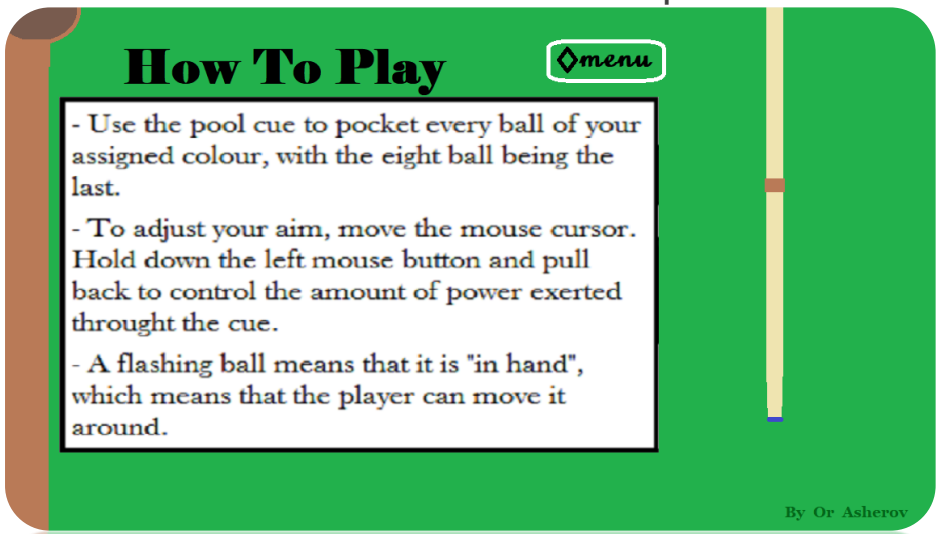


- לחצן " Exit " יוצא מהמשחק.

הפעולה שמציגה את מסך הפתיחה:

```
def draw_menu(game_in_progress, show_instructions):
    # get mouse position and click status
    mouse_x, mouse_y = pygame.mouse.get_pos()
    mouse_left = get_mouse_press()
    music=pygame.mixer.Sound("tap.wav")
    if show_instructions == False:
        # if mouse is hovering over a button, then highlight that button
        if mouse_x > 430 and mouse_x < 560 and mouse_y > 400 and mouse_y < 450: # start game
            pygame.mixer.Sound.play(music)
            myimage = pygame.image.load("Images/chose2.png")
            if mouse_left == True:
                game_in_progress = True
                pygame.mouse.set_pos([250,300]) # set mouse to proper starting position
        elif mouse_x > 680 and mouse_x < 870 and mouse_y > 240 and mouse_y < 290:   # show instructions
            myimage = pygame.image.load("Images/chose3.png")
            pygame.mixer.Sound.play(music)
            if mouse_left == True:
                show_instructions = True
        elif mouse_x > 215 and mouse_x < 330 and mouse_y > 270 and mouse_y < 315:   # exit game
            myimage = pygame.image.load("Images/chose1.png")
            pygame.mixer.Sound.play(music)
            if mouse_left == True:
                pygame.quit()
        else:
            myimage = pygame.image.load("Images/main.png")
    else: # show the instructions
        if mouse_x > 670 and mouse_x < 820 and mouse_y > 35 and mouse_y < 90:     # menu button
            myimage = pygame.image.load("Images/intro_selected.png")
            pygame.mixer.Sound.play(music)
            if mouse_left == True:   #back to main menu
                show_instructions = False
        else:
            myimage = pygame.image.load("Images/intro.png")
    # load and draw the menu
    imagerect = myimage.get_rect()
    screen.fill(BLACK)
    screen.blit(myimage, imagerect)
    pygame.display.flip()
    return game_in_progress, show_instructions
```

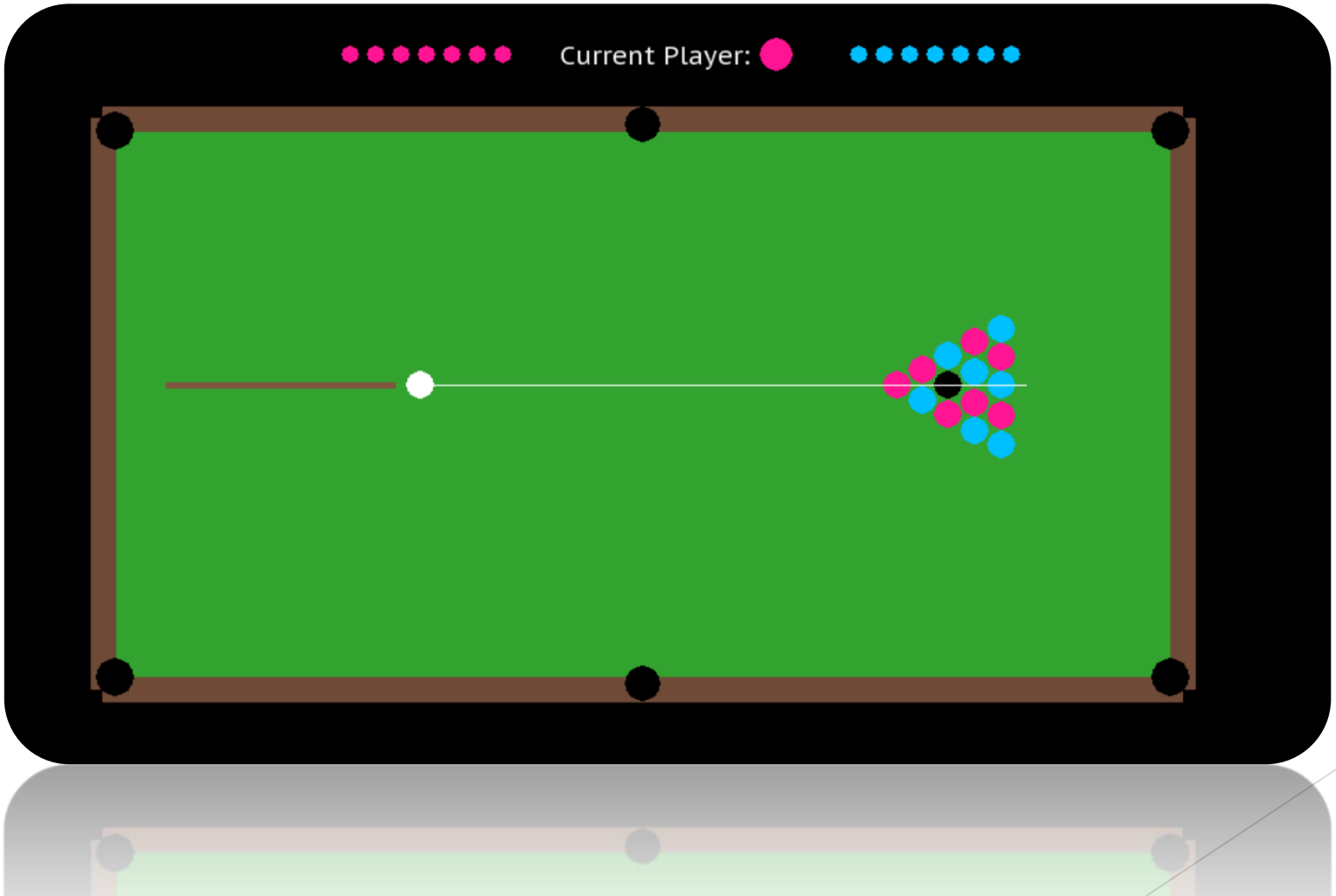
הוראות המשחק - "Instructions"



- כאשר לוחצים על כפתור זה נפתח מסך עם הוראות המשחק:
- השתמש במקל על מנת להכניס את כל הכדורים מהצבע שלך לחורים, כאשר הכדור השחור נכנס אחרון.
- לחץ על הכפתור השמאלי בעכבר וכך תכוון את כיוון הכדור הלבן. הרחק את המקל מהכדור הלבן על מנת לשלוט בעוצמת הפגיעה של המקל בכדור הלבן.
- כדור לבן "מהבהב" אומר שניתן להזיז את הכדור "עם היד", כלומר אתה יכול למקם את הכדור הלבן במקום בו תבחר.
- כל עוד העכבר נמצא על כפתור ה "Menu" נשמע שוב הצליל והכפתור משנה את צבעו (כמו במסך הפתיחה). אם לוחצים על כפתור זה חוזרים למסך הפתיחה.

התחלת המשחק - "Start"

- כשלוחצים על כפתור זה נפתח מסך המשחק:



9

- מיקמתי כל כדור על המסך (לכל כדור יש שיעור X ושיעור Y משלו, באמצעות Pygame).
- השחקן הראשון שמשחק הוא השחקן שלו הצבע הורוד ולאחר מכן השחקן שלו הצבע הכחול וכן הלאה.
- העכבר ממוקם במיקום אותו קבעתי (כך שהכדור הלבן ינוע ישר לעבר ראש הפירמידה של הכדורים).
- ישנם 7 כדורים ורודים, 7 כחולים, כדור לבן וכדור שחור.
- מעל שולחן הביליארד כתוב תור מי מבין השחקנים (ורוד/כחול) ואת כמות הכדורים שנותרה משני הצבעים .
- כאשר המקל פוגע בכדור הלבן נשמע צליל .
- כאשר הכדורים פוגעים זה בזה נשמע צליל .
- כאשר אחד מהכדורים על השולחן נכנס לחור נשמע צליל . אם אחד הכדורים שנכנס הוא כחול/ורוד, הכמות מאותו צבע (שמופיעה מעל השולחן) תפחת (בכמות הכדורים שנכנסו לחור).
- אם הכדור השחור נכנס על ידי שחקן כאשר עוד נותרו לו כדורים מצבעו, שחקן זה יפסיד. אם שחקן זה הכניס את כל כדוריו ולאחר מכן גם את הכדור השחור, הוא ינצח.

פעולות מרכזיות ליצירת האובייקטים במשחק באמצעות pygame

```
def draw_static_objects():
    # Draw the playing surface
    pygame.draw.rect(screen, GREEN, [80, 90, 850, 450], 0)   #pygame.draw.rect(screen, color, (x,y,width,height), thickness)
    # Draw the pool table border
    pygame.draw.rect(screen, DARK_BROWN, [80, 90, 850, 450], 20)  #pygame.draw.rect(screen, color, (x,y,width,height), thickness)
    # Draw the pockets
    #circle(Surface, color, pos, radius, width=0)
    pygame.draw.circle(screen, BLACK, (90, 100), 15, 0)      # top left
    pygame.draw.circle(screen, BLACK, (505, 95), 14, 0)     # top middle
    pygame.draw.circle(screen, BLACK, (920, 100), 15, 0)     # top right
    pygame.draw.circle(screen, BLACK, (90, 530), 15, 0)     # bottom left
    pygame.draw.circle(screen, BLACK, (505, 535), 14, 0)    # bottom middle
    pygame.draw.circle(screen, BLACK, (920, 530), 15, 0)    # bottom right
```

הפעולה שיוצרת
את שולחן
הביליארד והחורים:

```
def draw_scoreboard(current_player_turn):
    num_of_pink_left = 0
    num_of_blue_left = 0
    # get the number of pink and blue balls
    for ball_instance in Ball:
        if ball_instance.pocketed == False:
            if ball_instance.color == "Blue":
                num_of_blue_left += 1
            elif ball_instance.color == "Pink":
                num_of_pink_left += 1
    # draw the pink balls left
    for n in range(num_of_pink_left):
        pygame.draw.circle(screen, PINK, (275+n*20, 40), 7, 0)
    # draw the blue balls left
    for n in range(num_of_blue_left):
        pygame.draw.circle(screen, BLUE, (675+n*20, 40), 7, 0)
    # Draw an indicator, showing which player's turn it is
    draw_text("Current Player:", 440, 30, GREY, 20)
    if current_player_turn == "Pink":
        pygame.draw.circle(screen, PINK, (610, 40), 13, 0)
    elif current_player_turn == "Blue":
        pygame.draw.circle(screen, BLUE, (610, 40), 13, 0)
```

הפעולה שיוצרת
את לוח הנקודות
ומודיעה על תורות:

```
def draw_balls_list(balls_list):
    for x in xrange(len(balls_list)):
        if balls_list[x].pocketed == False:
            if balls_list[x].color == "Pink":
                pygame.draw.circle(screen, PINK, (int(balls_list[x].x), int(balls_list[x].y)), 11, 0)
            else:
                pygame.draw.circle(screen, BLUE, (int(balls_list[x].x), int(balls_list[x].y)), 11, 0)



def draw_balls():
    # draw the cue and eight balls
    if cue_ball.pocketed == False:
        pygame.draw.circle(screen, WHITE, (int(cue_ball.x), int(cue_ball.y)), 11, 0)

    if eight_ball.pocketed == False:
        pygame.draw.circle(screen, BLACK, (int(eight_ball.x), int(eight_ball.y)), 11, 0)

    # draw pink balls
    draw_balls_list(pink_balls)

    # draw blue balls
    draw_balls_list(blue_balls)
```

הפעולות שמציירות את כדורי הביליארד:

פעולות חישוביות במשחק באמצעות ספריית math

```
def angle_to_coordinates(angle, x, y):
    # get the x value by using the cosine function
    x = 1 * math.cos(math.radians(angle))
    # get the y value by using the sine function
    y = 1 * math.sin(math.radians(angle))
    return x, y

def get_angle(object1_x, object1_y, object2_x, object2_y):
    difference_of_x = object1_x - object2_x
    difference_of_y = object1_y - object2_y
    radians = math.atan2(difference_of_y, difference_of_x)
    radians %= 2*math.pi
    angle = math.degrees(radians)
    return angle

# Get the distance between two points
def get_distance(point1_x, point1_y, point2_x, point2_y):
    distance = math.sqrt((point1_x - point2_x)**2 + (point1_y - point2_y)**2)
    return distance

def convert_polar_coordinates_to_cartesian(x, y, angle, length):
    x += length * math.cos(math.radians(angle))
    y += length * math.sin(math.radians(angle))
    return x, y
```

הפעולה מחשבת את זווית הפגיעה
של שני האובייקטים ומחזירה אותה:

הפעולה מחשבת ומחזירה את המרחק
בין שתי נקודות (שני האובייקטים):

הפעולה מחשבת ומחזירה
את השיעורים (X ו Y)החדשים:

```python
def ball_to_cushion_collision(ball_direction, ball_speed, ball_x, ball_y, ball_in_contact):
    ball_hit_cushion = False
    if ball_in_contact == False:
        # hit the top cushion
        if ball_y < 113:
            ball_hit_cushion = True
            ball_direction = 360 - ball_direction

        # hit the bottom cushion
        if ball_y > 520:
            ball_hit_cushion = True
            ball_direction = 360 - ball_direction

        # hit the left cushion
        if ball_x < 100:
            ball_hit_cushion = True
            if ball_direction > 180 and ball_direction < 270: # ball incoming from the bottom
                ball_direction = 540 - ball_direction
            elif ball_direction > 90 and ball_direction < 180:                # ball coming from the top
                ball_direction = 180 - ball_direction
            elif ball_direction == 180: # direct hit
                ball_direction = 180

        # hit the right cushion
        if ball_x > 910:
            ball_hit_cushion = True
            if ball_direction > 270 and ball_direction < 360: # ball incoming from the bottom
                ball_direction = 540- ball_direction
            elif ball_direction > 0 and ball_direction < 90:                # ball coming from the top
                ball_direction = 180 - ball_direction
            elif ball_direction == 0: # direct hit
                ball_direction = 0
    else:
        ball_in_contact = False
    return ball_direction, ball_speed, ball_in_contact
```

הפעולה מחשבת את כיוון הכדור לאחר פגיעתו בדפנות השולחן:

מקורות עזר

- חישוב הזווית בין שתי נקודות (שימוש ב atan2 מספריית math):

http://math.stackexchange.com/questions/1201337/finding-the-angle-between-two-points

- מעבר מהצגה קוטבית להצגה קרטזית:

https://www.mathsisfun.com/polar-cartesian-coordinates.html

- הגדרת מחלקת Ball() כאיטרבלית:

http://stackoverflow.com/questions/739882/iterating-over-object-instances-of-a-given-class-in-python

- Font בו השמשתי:

www.fontsquirrel.com/fonts/PT-Sans