

## מבוא לספריית Arcade - הקמת סביבת המשחק - משימת פיתוח

### הקדמה

השלמתם צעד ענק! ביחידה האחרונה צללנו לעומק ספריית Arcade וגילינו איך באמת בונים עולמות: למדנו ליצור ולנהל אובייקטים (Sprites) והכרנו את 'הבסיס של המנוע הגרפי' שמניע כל משחק:

- מחלקת arcade.View
- setup: שמכין את זירת המשחק.
- on\_draw: שמצייר לנו את הקסם על המסך.

עכשיו, הגיע הזמן להעביר הילוך. זוכרים את הפק-מן שרץ לנו עד עכשיו בתוך הקונסול? תשכחו מזה! אנחנו יוצאים מהטקסט האפור ועוברים למשחק אמיתי, חי ובוועט. בשלבים הבאים נהפוך את הקוד שלכם לחוויה גרפית מלאה, עד שיהיה לכם פק-מן משלכם שתוכלו לשחק בו במחשב בדיוק כמו במכונות המשחק הקלאסיות. מוכנים להתחיל לקודד? :

### תרגיל חימום

לפני שאנחנו ניגשים לבניית מנוע המשחק והלוגיקה המורכבת, אנחנו צריכים להחליט איך המשחק שלנו ייראה. ב-Arcade. המראה של כל אובייקט נקבע על ידי ה-Texture (טקסטורה) שלו. בשלב זה, האחריות על מציאת הדרך הנכונה לעצב את הדמויות היא עליכם. עליכם לחקור את התיעוד של הספרייה ולהבין אילו פונקציות יאפשרו לכם ליצור את הצורות הבאות:

#### מה עליכם להגדיר?

1. Pacman: עיגול צהוב.
2. Ghost: עיגול אדום.
3. Coin: עיגול צהוב קטן.
4. Wall: ריבוע כחול.

איך מוצאים את הפונקציות המתאימות? היכנסו [לקישור הבא](#) של ספריית Arcade. חפשו שם פונקציות שמתחילות במילה \_make ויוצרות טקסטורות של צורות גיאומטריות (כמו עיגול או ריבוע).

כדי שהמשחק יידע איך להתייחס לדמויות שלנו במרחב, אנחנו צריכים להוסיף לכל מחלקה (Class) שני מאפיינים חדשים בתוך ה-`__init__`:

- width (רוחב)
- height (גובה)

שימו לב לאתגר: אל תמציאו סתם מספרים! חשבו מאיפה בתוך הקוד, או מאיזו תכונה של הטקסטורה שמצאתם - אפשר להפיק את הערכים האלו. האם יש קשר בין גודל הטקסטורה שיצרתם לבין הרוחב והגובה של הדמות?



### אתחול מחלקת PacmanGame

המחלקה PacmanGame מהווה את המנוע של המשחק. במחלקה זו נגדיר את כל האופרציה אשר קשורה בהרצת המשחק. המחלקה יורשת מ-`arcade.View`. מחלקת View בארקיייד מייצגת מסך במשחק. כל View יודע:

- לצייר את עצמו
- להתעדכן בכל פריים
- לקבל קלט מהשחקן

**מאפייני המחלקה כוללים:** רשימת עצמים, אובייקט שחקן, מצב המשחק (האם ניצחנו או לא) ואת הצבע של הרקע (שחור).

### מתודת הבנאי - `__init__`

המתודה שמאתחלת את המחלקה עצמה. נשים לב שמתודה זו לא בונה את המשחק, אלא רק מאתחלת אובייקט חדש של המשחק. המתודה תאתחל את השדות הבאים:

- `self.wall_list` - רשימת Sprites המייצגים את הקירות במשחק.
- `self.coin_list` - רשימת Sprites המייצגים את המטבעות במשחק.
- `self.ghost_list` - רשימת Sprites המייצגים את הרוחות במשחק.
- `self.player_list` - רשימת Sprites המייצגים את השחקנים במשחק.
- `self.player` - בהמשך ייצג את השחקן, כעת נאתחל אותו כ-`None`.
- `self.game_over` - כעת נאתחל אותו כ-`False`.
- `Self.background_color` - יאותחל בצבע שחור בעזרת `arcade.color.BLACK`.
- `self.start_x` - מייצג את נקודת ההתחלה של השחקן בציר ה-x, נאתחל כ-0.
- `self.start_y` - מייצג את נקודת ההתחלה של השחקן בציר ה-y, נאתחל כ-0.

## מתודת setup

מקבלת את self כארגומנט ומאתחלת את המשחק עצמו. במתודה זו אנו נעבור על מפת המשחק ונוסיף את האובייקטים השונים למחלקות הרלוונטיות. המתודה מופעלת פעם אחת לפני שהמשחק מתחיל, ומבצעת את הפעולות הבאות:

1. יצירת ה-SpriteList חדשות (כמו ב-init).
  2. מוודא איפוס של דגל המשחק שלנו (שינוי ל-False).
  3. בניית העולם לפי המפה שלנו. נעבור על המפה בצורה הבאה:
- ```
for row_idx, row in enumerate(LEVEL_MAP):
    for col_idx, cell in enumerate(row):
        x = col_idx * TILE_SIZE + TILE_SIZE / 2
        y = (rows - row_idx - 1) * TILE_SIZE + TILE_SIZE / 2
```
4. הוספת אובייקטים לרשימות הרלוונטיות.

## מתודת on\_draw

מקבלת את self כארגומנט, ומבצעת את ציור המשחק למסך. המתודה תהיה אחראית על הציורים הבאים:

1. קירות.
2. רוחות.
3. מטבעות.
4. שחקן.
5. טקסט בצד שמאל למעלה שמתאר כמה נקודות יש לשחקן.
6. טקסט בצד שמאל למעלה שמתאר כמה חיים יש לשחקן.
7. במידה והפסדנו - טקסט במרכז הלוח.

## דגשים



- על מנת לבצע עבודה נכונה - עבדו עם קבועים. הגדירו את הקבועים הבאים בתחילת המשחק:
  1. WINDOW\_WIDTH - אחראי על רוחב החלון, מאותחל כ-800.
  2. WINDOW\_HEIGHT - אחראי על אורך החלון, מאותחל כ-600.
  3. WINDOW\_TITLE - הכיתוב שיופיע בראש החלון.
  4. TILE\_SIZE - גודל האריח במשחק (כמה פיקסלים צריך כל ציור), מאותחל כ-32.
  5. LEVEL\_MAP - רשימה של מחרוזות, תכיל את מפת המבוך.
- Sprite היא יחידת הבסיס בספריית Arcade, שבעזרתה אנו יוצרים דמויות וצורות במשחק.

מכיוון שלא ביצענו `from arcade import Sprite`, עלינו לפנות אליה באמצעות השם המלא `arcade.Sprite`.



על מנת להריץ את המשחק, יש ליצור תחילה חלון משחק באמצעות המחלקה `Window` מספריית `Arcade`. לאחר מכן יש ליצור מופע של מחלקת המשחק (`PacmanGame`) ולבצע אתחול מלא של רכיבי המשחק באמצעות המתודה `setup()`. בסיום, יש להציג את המסך הראשי של המשחק באמצעות `show_view()`, ואז להתחיל את לולאת המשחק של `Arcade` באמצעות הקריאה `arcade.run()`. לולאה זו מפעילה ברצף את כל מחזורי המשחק.

### הפעולות חייבות להופיע בסדר זה, כיוון שכל שלב תלוי בקודמים לו:

1. יצירת חלון – ממשק גרפי שבו המשחק יוצג.
2. יצירת אובייקט המשחק – שמנהל את הלוגיקה, המפה והספרייטים.
3. אתחול עולם המשחק (`setup`) – בנייה מחדש של קירות, שחקן, מטבעות ורוחות.
4. הצגת ה-`View` – מחייב הצבת המשחק בחלון.
5. הפעלת לולאת המשחק (`arcade.run`) – מפעילה את המנוע שמריץ את המשחק בפועל.