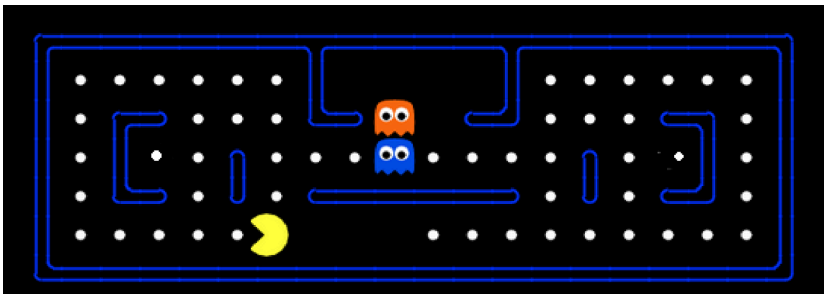
# תרגיל בית 1- פתרון בעיות על ידי חיפוש

## מבוא

מטרת התרגיל היא להתנסות בפתרון אחת הגרסאות של משחק pacman בעזרת שיטות חיפוש שונות, אותן למדנו בקורס.



עקרון המשחק הוא ש-pacman (הדמות הצהובה) צריכה "לאכול" את כל האסימונים (הנקודות הלבנות בתמונה למעלה), מבלי להיאכל על ידי הרוחות רפאים שיש על הלוח

לוח המשחק מורכב מ – NxM משבצות ומכיל 4 סוגים של אוביקטים:

- רוחות רפאים
- אסימונים
- קיר
- שחקן (ה-pacman)

## הבעיה

כקלט תקבלו את המצב ההתחלתי של המשחק. זוהי מטריצה (game) בגודל NxM שמייצגת את לוח המשחק, הכניסות במטריצה מסמנות את המצב ההתחלתי של האוביקטים בו. השחקן יכול לזוז ימינה, שמאלה, למעלה ולמטה, בתנאי שבמשבצת אליה הוא זז אין קיר.

כמו שהוזכר, מטרת המשחק היא להוביל את השחקן על גבי לוח המשחק, כך שיאכל את כל האסימונים, מבלי להיאכל על ידי אף אחת מהמפלצות הנעות על גבי הלוח. המשחק יכול להגמר ב 2 אופנים:

- **הצלחה:** השחקן אכל את כל האסימונים
- **כשלון:** אחת המפלצות "אוכלת" את השחקן. יכול להגרם על ידי כניסה של השחקן למשבצת בה יש רוח רפאים או כניסה של רוח רפאים אל המשבצת של השחקן

## תיאור המשימה

לתרגיל זה מצורף קוד הממש אלגוריתמי חיפוש שלמדנו. עליכם לממש את המחלקה המייצגת משחק Pacman כך שניתן יהיה לפתור משחק נתון על ידי שימוש בקוד החיפוש המצורף.

קובץ ex1.py המצורף מכיל את חתימות הפונקציות שעליכם לממש. (ניתן כמובן גם למשש פונקציות נוספות):

1. פונקציית successor המקבלת מצב ומחזירה tuple הכולל את כל הפעולות המותרות מאותו המצב
2. פונקציית result המקבלת מצב ופעולה ומחזירה מצב חדש הנוצר כתוצאה מהפעלת פעולה נתונה במצב נתון
3. פונקציית goal_test המקבלת מצב ומחזירה true אם המצב הנתון הוא מצב מטרה (מה שמוגדר כהצלחה בסעיף קודם), ו-false אחרת
4. פונקציית היוריסטיקה (h) המקבלת מצב ומחזירה ומחזירה את העלות המשוערת להגעה מהמצב הנתון למטרה. שימו לב כי טיב ההיוריסטיקה ישפיע על הביצועים של אלגוריתמי החיפוש.
5. עליכם גם לכתוב את מספרי תעודת זהות שלכם במשתנה id מחוץ למחלקה

<u>אין לשנות את חתימות הפונקציות כלל!</u>

## ייצוג קלט

הקלט שמתאר את הבעיה מועבר לפונקציה

```
create_pacman_problem(game)
```

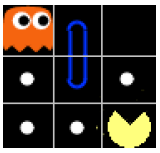
כאשר:

- 99 – קיר
- 88 – שחקן נאכל ע"י רוח רפאים
- 77 – שחקן (פאקמן)
- 51 – רוח רפאים **אדומה** במשבצת עם גלולה
- 50 – רוח רפאים **אדומה** במשבצת ללא גלולה
- 41 - רוח רפאים **ירוקה** במשבצת עם גלולה
- 40 - רוח רפאים **ירוקה** במשבצת ללא גלולה
- 31 - רוח רפאים **צהובה** במשבצת עם גלולה
- 30 - רוח רפאים **צהובה** במשבצת ללא גלולה
- 21 - רוח רפאים **כחולה** במשבצת עם גלולה
- 20 - רוח רפאים **כחולה** במשבצת ללא גלולה
- 11 – משבצת רגילה עם גלולה
- 10 – משבצת ריקה

לדוגמא הקריאה:

```
create_pacman_problem (
(51, 99, 10),
(11, 99, 11),
(11, 11, 77))
```

מתארת את המשחק הבא:



3 על 3, ונמצאים בו 2 קירות, 4 משבצות אסימונים, משבצת ריקה, רוח רפאים אחת, והשחקן (ה-pacman).

## אופן וחוקיות התזוזה

לרשותכם 4 פעולות של השחקן:

- תזוזה ימינה – מיצוגת ע"י מחרוזת "R"
- תזוזה למטה – מיצוגת ע"י מחרוזת "D"
- תזוזה שמאלה – מיצוגת ע"י מחרוזת "L"
- תזוזה למעלה – מיצוגת ע"י מחרוזת "U"

כל פועלה כזאת גורמת לפאקמן לזוז משבצת אחת בכיוון הרצוי, חוץ מתזוזה לכיוון הקיר, שלא תשנה את מיקום השחקן. חוץ מזה, אחרי זה גם רוחות הרפאים זזות. תזוזת רוחות הרפאים מתבצעת לפי החוקיות הבאה:

- כל רוח זזה משבצת אחת לאחד מ-4 הכיוונים (למעלה, למטה, ימינה או שמאלה)
- סדר התזוזה: שחקן, אחריו רוח אדומה, אחרי זה כחולה, צהובה ולבסוף ירוקה
- כל רוח תזוז לכיוון הכי מקרב אותה לשחקן לפי מרחק מנהטן
- אם תזוזת הרוח ל-2 כיוונים מקרבת אותה לשחקן באותה מידה, אז סדר העדיפויות לתזוזה צריך להיות: ימינה, למטה, שמאלה ולבסוף למעלה.

- רוח לא יכולה להישאר במקום, אלא אם כן אין לה משבצת שאליה היא יכולה לעבור.
- רוח לא יכולה לעבור למשבצת שנמצאת בה רוח אחרת או קיר. (משמע: בשום מצב שתי רוחות לא יכולות להימצא באותה משבצת.)

שימו לב, שאת תנועת הרוחות יש לחשב באופן הזה בלבד ולא על פי כל אלגוריתם אחר. פתרונות שיוחזרו עבור לוחות בהן הרוחות נעו באופן אחר מהמתואר לעיל ייכשלו בבדיקת הבודק האוטומטי.

<u>הערה</u>: מצב כישלון במשחק (רוח אכלה את השחקן) אין להמשיך לפתח את המצבים על ענף זה.

## דגשים

- שימו לב שכדי לקבל את המצב מתוך קודקוד החיפוש (node), ניתן לגשת ל-node.state
- אתם יכולים לייצג מצב של הבעיה איך שאתם רוצים, אבל שימו לב שהקוד של החיפוש דורש שהמצב יהיה hashable , ולכן לא ניתן להשתמש ברשימה או במילון (או ב – tuple המכיל רשימה או מילון) על מנת ליצג מצב. ניתן לממש מחלקה המתארת מצב ויש לה פונקצית hash או להשתמש ב – tuple
- אמנם ייצוג המצב הוא בחירה שלכם אולם יש להקפיד על ייצוג הפעולות בדיוק כפי שכתוב – אותיות גדולות/קטנות, קווים תחתונים, ייצוג הפעולות נכון וכו. הבדיקה אוטומטית, ולכן אם תטעו כאן תקבלו ציון נמוך מאד, וחבל.

## בדיקת התרגיל

התרגיל המוגש ייבדק באופן אוטומטי, ולכן חשוב להקפיד על שמות מדויקים של קבצים, מחלקות ופעולות. הבדיקה האוטומטית תשתמש באלגוריתמי חיפוש שונים (A* וGBFS) על מנת לפתור אוסף קלטים בגדלים שונים. לאחר מכן הפתרון ייבדק צעד אחר צעד על מנת לוודא שהפתרון אכן תקין.

הקובץ ex1_check.py המצורף מריץ כל את אלגוריתמי החיפוש שנבדוק על מספר קלטים. קובץ זה לא מבצע בדיקת נכונות של הפתרון המוחזר, ולכן זוהי אחריות שלכם לוודא שהפתרונות שלכם נכונים.

הרצת הקובץ באמצעות הפקודה:

```
python3 ex1_check.py
```

## הסבר קצר על הקוד המצורף

הקוד מורכב מחמישה קבצי פייתון:

1. Ex1.py – קובץ שבו נעשית העבודה העיקרית שלכם, והקובץ היחיד שעליכם לשנות. אמור להכיל את ה-class של PacmanProblem המכיל את כל הפונקציות כפי שמתואר בסעיף "תיאור משימה". הקובץ כולל את חתימות של הפונקציות שעליכם לממש. **התרגיל יבדק עם קובץ ex1.py שלכם, ושאר הקבצים כפי שהם מופיעים במצבם המקורי ולכן אין טעם לשנות קבצים אחרים** (למעט הבעיות השונות שנבדוק עליהם את הקוד).
2. ex1_check.py – קובץ המכיל פונקציות מעטפת המנסות לפתור את הבעיה, ומכיל בעיה קטנה לדוגמא שאפשר לפתור. זה הקובץ שעליכם להריץ לבדיקת הפתרון שלכם.
3. Search.py – מכיל את אלגוריתמי החיפוש, ומכיל את מחלקת node.
4. שאר הקבצים – קבצי עזר

## הגשה

- תאריך ההגשה עד ה 5.2.24
- הגשה ביחידים בלבד
- את הת.ז. של המגיש יש לרשום במשתנה id בקובץ ex1.py וכן לצרף קובץ details.txt המכיל את שם ות.ז של המגיש.

- שאלות על התרגיל יש לשאול בפורום שאלות ותשובות יעודי ב"למדה". שאלות יענו אחת ליומיים - שלושה במרוכז.
- הגשת התרגיל תהיה דרך מערכת ה submit. יש להגיש רק את הקובץ ex1.py וקובץ details.txt. אין להגיש קבצי עזר המצורפים לתרגיל. אין להגיש קובץ בפורמט אחר (לדוגמא zip או rar)