

# **כיתה ז – תרגילי כיתה בגרסה האנגלית**

**סקראצ web-2 הגרסה האנגלית**

**תרגילי כיתה**

**פיתוח:**

**מורים מובילים לחטיבת הביניים**

**בהנחיית ד"ר דורון זוהר**

## תרגיל הסירות, גב' אלה לב, הגימנסיה הראלית, ראשון לציון

קבוצת מטיילים מגיעה מדי יום לשיט בסירות גומי בירקון. בסירה אחת נכנסים 8 אנשים. מחיר השכרה של סירה אחת הוא 50 ₪. יש לכתוב תסריט המקבל כקלט את מספר האנשים בקבוצה. על התסריט לחשב את מספר הסירות שיש לשכור עבור הקבוצה ואת המחיר שיש לשלם עבור השייט.

נבצע את הפתרון על פי השלבים הבאים:

- שלב 1 - דוגמאות קלט מייצגות.** נבחר מספר דוגמאות ונבחן לכל דוגמה יהיה הפלט.
- שלב 2 - פירוק המשימה לתת משימות.** ניתוח הבעיה ופירוקה למשימות קטנות יותר.
- שלב 3 - כתיבת האלגוריתם.** כתיבת אלגוריתם מילולי לפתרון הבעיה.
- שלב 4 - כתיבת התסריט.** תרגום האלגוריתם לתסריט בסקראצ'.
- שלב 5 - טבלת מעקב.** בדיקת התסריט בהרצה "יבשה" עבור קלטים מייצגים. שינוי ותיקונים במידת הצורך.

### **לפני פתרון הבעיה המרכזית נפתור את בעיה דומה אך פשוטה יותר:**

קבוצת מטיילים מגיעה מדי יום לשיט בסירות גומי בירקון. בסירה אחת יכולים להיכנס 8 אנשים. מפעילי המועדון מאפשרים רק לסירות **מלאות** לצאת לשייט. דמות החתול מבקשת לתכנת תסריט הבודק האם קבוצת מטיילים יכולה למלא בדיוק מספר שלם של סירות. לתשובה חיובית, בה מספר המטיילים מתחלק לשמונה ללא שארית, הדמות תאמר כמה סירות ימלאו חברי הקבוצה. בכל מקרה אחר תאמר הדמות לכמה מטיילים אין אפשרות לעלות על הסירות.

יש לכתוב תסריט המקבל כקלט את מספר המטיילים בקבוצה ומודיע האם יכולים לצאת לשייט ובכמה סירות, בכל מקרה אחר תאמר הדמות את מספר המטיילים שאינם ממלאים סירה.

נבצע את הפתרון בשלבים .

### **שלב 1 – דוגמאות קלט מייצגות**

קלט	הדמות תאמר
10	לא ניתן לצאת לשיט, מספר האנשים העודפים 2
27	לא ניתן לצאת לשיט, מספר האנשים העודפים 3
6	לא ניתן לצאת לשיט, מספר האנשים העודפים 6
8	ניתן לצאת לשיט, מספר הסירות 1
40	ניתן לצאת לשיט, מספר הסירות 5

## שלב 2 – פירוק המשימה לתת משימות

קבלת מספר אנשים שהגיעו לשייט (מהמשתמש)  
בדיקה האם מספר האנשים מתחלק ב-8 ללא שארית  
אם כן – טיפול מתאים ל - "כן"  
אחרת ( אם לא ) – טיפול מתאים ל - "לא"

## שלב 3 – כתיבת האלגוריתם ראשוני

יש לקבל את מספר האנשים שהגיעו לשייט ולהשים ערך זה במשתנה : `num_of_people`  
יש לבדוק האם מספר האנשים מתחלק ב-8 ללא שארית  
אם כן

יש לחשב את מספר הסירות  
הדמות מודיעה שאפשר לצאת לשייט ומודיעה מה מספר הסירות

אחרת

יש לחשב כמה אנשים עודפים יש .  
הדמות מודיעה שאי אפשר לצאת לשייט כי יש מספר אנשים עודפים ואומרת כמה  
עודפים יש.

שלב 3		שלב 4
1	איתחול	
2	num_of_people הגדרת המשתנה	
3	num_of_people קבלת ערך למשתנה	
4	num_of_people אם מתחלק ב-8 ללא שארית	
5	<ul style="list-style-type: none"> <li>The group can go on a boat "trip</li> <li>Number of boats is....."</li> </ul>	
6	אחרת	
7	<ul style="list-style-type: none"> <li>" The group can't go. You have extra ....."</li> </ul>	

### שלב 5 - טבלת מעקב

שורה	num_of_people	תנאי	פלט
1		$\text{num\_of\_people} \bmod 8 = 0$	בתסריט
2			1
3			2
4			How many
5	27		
6		שקר	
7			The group can't go.

## פתרון 2 הוספת אילוץ חדש:

נתייחס לבעייה שהוצגה בתרגיל הקודם.

כיצד ישתנה התסריט אם נאפשר לקבוצה **קטנה** המגיעה לשוט ומונה פחות מ- 8 אנשים

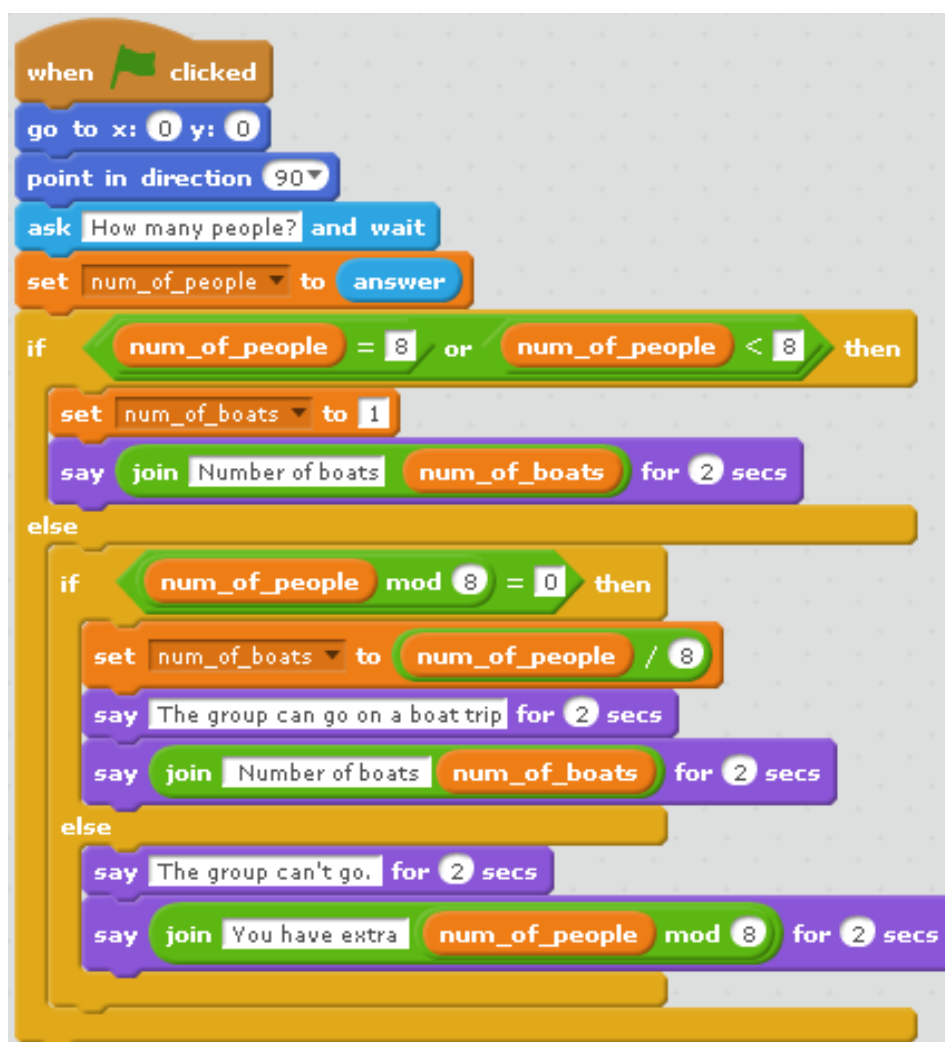
לקחת סירה, למרות שהקבוצה לא ממלאת סירה?

גם קבוצה של אנשים הנכנסים כולם לסירות, יכולה לצאת לשייט.

לקבוצות גדולות **מדי** עדיין לא ניתן לצאת לשייט בגלל שיש אנשים שצריכים לשבת בצד.

יש לפתור את הבעייה בשלבים כמו שראית בדוגמה הקודמת.

דוגמת פתרון:



```
when clicked
  go to x: 0 y: 0
  point in direction 90
  ask How many people? and wait
  set num_of_people to answer
  if num_of_people = 8 or num_of_people < 8 then
    set num_of_boats to 1
    say join Number of boats num_of_boats for 2 secs
  else
    if num_of_people mod 8 = 0 then
      set num_of_boats to num_of_people / 8
      say The group can go on a boat trip for 2 secs
      say join Number of boats num_of_boats for 2 secs
    else
      say The group can't go. for 2 secs
      say join You have extra num_of_people mod 8 for 2 secs
```

### פתרון 3 : שוב, הוספת אילוץ חדש:

נתייחס לבעייה שהוצגה בתרגיל הקודם.

קעת הבשילו התנאים לטפל במשימה המורכבת של מציאת מספר סירות גם עבור קבוצה גדולה של מטיילים.

יש לשים לב שמספר הסירות הנדרש אינו מתחלק בדיוק לשמונה מטיילים בסירה. במקרה כזה יש להעזר בסירה נוספת כך שינתן מענה לכל המטיילים בקבוצה. שימו לב כי תוצאת החלוקה יכולה להיות עשרונית (שבר).

רמז: אם נוריד ממספר המטיילים הכולל את מספר האנשים שצריכים לשבת בצד, נקבל מספר שמתחלק ב-8. אם אכן נחלק את התוצאה הזו ב-8, נקבל את מספר הסירות המלאות (מספר שלם) ואז נוסיף עוד סירה.

יש לפתור את הבעייה בשלבים כמו שראית בדוגמאות קודמות.

דוגמת פתרון:

```
when green flag clicked
  go to x: 0 y: 0
  point in direction 90
  ask "How many people?" and wait
  set num_of_people to answer
  if (num_of_people = 8) or (num_of_people < 8) then
    set num_of_boats to 1
  else
    if (num_of_people mod 8 = 0) then
      set num_of_boats to (num_of_people / 8)
    else
      set num_of_boats to ((num_of_people - (num_of_people mod 8)) / 8) + 1
  end if
  say join "Number of boats " num_of_boats for 2 secs
```

#### פתרון 4 : נחזור לבעייה המקורית, כולל עלות השייט:

כעת נשלים את הפתרון ונחשב את עלות השייט.

דוגמת פתרון:

```
when clicked
  go to x: 0 y: 0
  point in direction 90
  ask How many people? and wait
  set num_of_people to answer
  if num_of_people = 8 or num_of_people < 8 then
    set num_of_boats to 1
  else
    if num_of_people mod 8 = 0 then
      set num_of_boats to num_of_people / 8
    else
      set num_of_boats to (num_of_people - num_of_people mod 8) / 8 + 1
  say join Number of boats num_of_boats for 2 secs
  set total_price to num_of_boats * 50
  say join Total price of the trip is total_price for 2 secs
```



## משחק איקס-עיגול (Tic-Tac-Toe), מר מני עבודי

### המשימה

יש לפתח את המשחק הפופולרי איקס עיגול או בשמו הלועזי Tic-Tac-Toe, בפורמט של משחק בין שני שחקנים.

שאלות ותשובות לדיון בכיתה לפני ביצוע התרגיל:

- **ש:** כמה דמויות נצטרך להגדיר למשחק?  
**ת:** הדמויות הבסיסיות הן X ו-O. בהתאם לעיצוב המשחק, ניתן להגדיר דמויות נוספות במשחק, לדוגמה: לוח המשחק
- **ש:** כיצד יקבע מי יתחיל את המשחק?  
**ת:** ניתן לקבוע מי הדמות שתתחיל את המשחק במספר אופנים: קליטה של זהות השחקן שיתחיל, הגרלה וכיו"ב.
- **ש:** באיזה מבנה נתונים נייצג את לוח המשחק?  
**ת:** לוח משחק סטנדרטי מורכב ממטריצה בגודל 3x3. מכיוון שאין מימוש של מבנה נתונים מסוג מערך ב scratch 2.0, ניתן לייצג את הלוח במספר אופנים: משתנה או רשימה (list).
- **ש:** אילו ערכים באים לידי ביטוי בכל תא בלוח המשחק?  
**ת:** כל תא בלוח המשחק יכול להכיל אחד מבין הערכים הבאים: ריק או X או O.
- **ש:** כיצד נזכור את מצב לוח המשחק?  
**ת:** בהתאם למבנה הנתונים שנבחר ללוח המשחק, נצטרך להחליט זאת.
- **ש:** מהו מצב של ניצחון? מהו מצב של תיקו? כיצד נזהה מצבים אילו?  
**ת:** מצב ניצחון הוא מצב שבו שחקן מסוים (X או O) מייצר רצף של שורה או עמודה או אלכסון. מצב תיקו הוא מצב בו אף שחקן לא הצליח לייצר מצב של ניצחון ולא נותר עוד תא פנוי בלוח המשחק.



## משימה 1 – בחירת השחקן הפותח

ניתן לבחור את השחקן הפותח במספר אופנים.

הגרלת בחירת השחקן הפותח:

מנגנון רנדומליות (אקראיות) קיים ב scratch בתפריט operators (מפעילים). יש להשתמש

במנגנון זה בכדי לבחור את השחקן הפותח: שחקן X או שחקן O.

פתחו ויישמו תסריט לקביעת השחקן הפותח. הניחו כי למשחק יש שני שחקנים בלבד.

### פתרון מוצע



תסריט זה מהווה יחידת עבודה עצמאית אשר ניתן להשתמש בה בתסריטים נוספים.

במקום לכתוב את רצף ההוראות בכל פעם מחדש, נמדל את התסריט ונכתוב פעולה לבחירת השחקן הפותח באמצעות תפריט more blocks (לבנים נוספות) ונגדיר פעולה שאינה מקבלת פרמטרים נוספים ומחזירה (באמצעות מסרים) את זהות השחקן הפותח באופן הבא<sup>1</sup>:



<sup>1</sup> במימוש הנוכחי, השחקנים (שחקן X ושחקן O) מיוצגים כל אחד כדמות נפרדת. פתיחת המשחק נעשית מתוך תסריט של לוח המשחק, המיוצג כבמה. לאחר בחירת השחקן הפותח, מסר מתאים נשלח לדמות המייצגת את השחקן הפותח.

## משימה 2 – אתחול לוח המשחק

לוח המשחק הבסיסי הוא מבנה לוח בגודל  $3 \times 3$ . ב scratch לא קיים מבנה נתונים של מערך דו-מימדי ולכן לפנינו ניצבות מספר אפשרויות: ניתן לייצג את הלוח במספר אופנים: משתנה או רשימה (list).

מבנה נתונים רשימה (list) הוא מבנה נתונים נוח לשימוש לצורך עדכון מצב הלוח בכל שלב במשחק.

מבנה נתונים רשימה דינאמי הכולל מספר תכונות ומתודות פנימיות, לדוגמה:

הכנסה/עדכון/מחיקה של איבר ברשימה במיקום מסויים.

כאמור, כל תא בלוח המשחק מכיל אחד מ- 3 המצבים הבאים:

- תא ריק (מיוצג ע"י הערך #),

- תא שנבחר ע"י שחקן X (מיוצג ע"י הערך X)

- ותא שנבחר ע"י שחקן O (מיוצג ע"י הערך O).

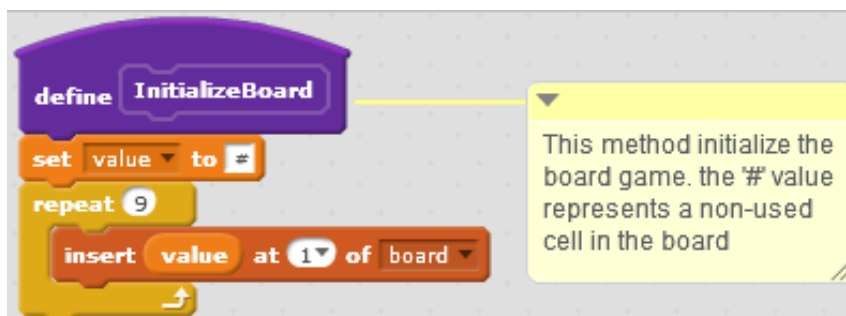
פתחו ויישמו תסריט לאתחול לוח המשחק. הניחו כי לוח המשחק הוא בגודל  $3 \times 3$ .

### פתרון מוצע



כמו בשלב הקודם, במקום לכתוב את רצף הוראות בכל פעם מחדש, נמדל את התסריט ונכתוב פעולה לאתחול לוח המשחק באמצעות תפריט more blocks ונגדיר פעולה שאינה

מקבלת פרמטרים נוספים ושאינה מחזירה ערך<sup>2</sup> באופן הבא:



<sup>2</sup> במימוש הנ"ל לוח המשחק מיוצג באמצעות במה (stage). לוח המשחק, המיוצג כרשימה בשם board, מתפקד כמבנה נתונים גלובלי לכל הדמויות.

## הסבר על מבנה הנתונים board (לוח המשחק)

לוח המשחק (board) נדרש לייצג מטריצה בגודל  $3 \times 3$ . כיצד ניתן לייצג את לוח המשחק כרשימה?

בייצוג הנוכחי, כל איבר ברשימה מייצג תא (cell) בלוח המשחק, כך שהאיבר ה-1 ברשימה מייצג את התא בשורה ה-1 ובעמודה ה-1 בלוח המשחק. האיבר ה-2 ברשימה מייצג את התא בשורה ה-1 ובעמודה ה-2 בלוח המשחק וכך הלאה. להמחשה, הסתכלו על לוח המשחק הבא<sup>3</sup>:

X	O	O
X	O	O
O	X	X

board	
1	X
2	O
3	O
4	X
5	O
6	O
7	O
8	X
9	X

+ length: 9

לוח המשחק, המיוצג כרשימה, יראה כך:

### משימה 3 – פתיחת המשחק

מה כוללת פתיחת המשחק?

מתוך שתי המשימות הקודמות, אנו מבחינים שיש לפחות שתי פעולות שמתבצעות בפתיחת המשחק:

א. אתחול לוח המשחק

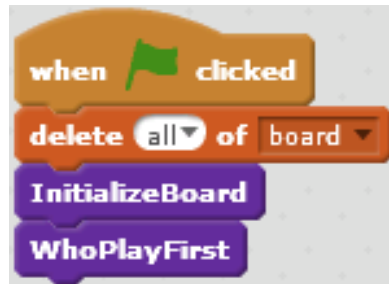
ב. בחירת השחקן הפותח

פתחו וממשו תסריט לפתיחת המשחק.

<sup>3</sup> במקרה זה, שחקן O היה הראשון לפתוח

## פתרון מוצע<sup>4</sup>

ניעזר בשתי הפעולות שהגדרנו בשלבים הקודמים ונשלבם בתסריט האחראי על פתיחת המשחק.



### משימה 4 – עדכון מצב לוח משחק

בכל מהלך של שחקן (X או O) יש לעדכן את מצב לוח המשחק. בהתאם למהלך, יש לבחון את מצב המשחק (ניצחון או תיקו). במשימה זו נתרכז בעדכון לוח המשחק.

כיצד נעדכן את לוח המשחק?

בסביבת scratch ישנם מספר רחב של מגנונים לפתרון הבעיה. בהתאם למימוש הנוכחי<sup>5</sup>,

נממש את עדכון הלוח באמצעות דמויות, המעבירות מסר לעדכון לוח המשחק.

כל עוד תא מסויים אינו בשימוש הדמות של אותו תא תוצג על גבי הלוח. במידה ושחקן מסויים

מעוניין לשבץ את עצמו בתא זה, הוא ילחץ עליו. הלחיצה תפעיל אירוע מסוג

" when this sprite clicked ", אשר תגרור רצף של הוראות:

- הסתרה (hide) של הדמות המייצגת תא.

- עדכון של הלוח בהתאם לשחקן הנוכחי.

- שיבוט של השחקן הנוכחי ומיקומו בתא המבוקש.

- בדיקת מצב משחק (ניצחון/תיקו/המשך משחק).

- מעבר לשחקן הבא.

פתחו וממשו תסריט לעדכון לוח המשחק, בהינתן תא מסויים בלוח המשחק ושחקן כלשהו<sup>6</sup>.

<sup>4</sup> במימוש הנוכחי, השחקנים (שחקן X ושחקן O) מיוצגים כל אחד כדמות נפרדת. פתיחת המשחק ניעשת מתוך תסריט של לוח המשחק, המיוצג כבמה.

<sup>5</sup> לוח המשחק מיוצג באמצעות רשימה ונראה למשתמש כרקע של במה.

<sup>6</sup> חשבו האם בפעולה לעדכון לוח המשחק נדרש פרמטרים, כלומר מהו המידע הנדרש לביצוע פעולת עדכון הלוח במימוש הנוכחי?

```

when this sprite clicked
  set position to 1
  UpdateBoard position player
  broadcast check_if_player_wins

when clicked
  show
  
```

```

define UpdateBoard position player
  if item position of board = # then
    replace item position of board with player
    hide
    broadcast check_if_player_wins
    if player = X then
      broadcast show_playerX_c1
      set player to O
    else
      broadcast show_playerO_c1
      set player to X
    else
      say Choose another cell! for 2 secs
  
```

This method updates the board with player according to its position

ברמת שחקן<sup>7</sup>

```

when I receive show_playerX_c1
  create clone of myself
  go to x: -111 y: 45
  
```

<sup>7</sup> שחקן X, למשל

## משימה 5 – בדיקת מצב משחק

ישנם שלושה מצבים אפשריים למשחק: ניצחון (של אחד מהשחקנים), תיקו או המשך משחק, כלומר מצב שאינו ניצחון או תיקו.

מהו מצב ניצחון?

מצב ניצחון הוא מצב שבו שחקן מסוים (X או O) מייצר רצף של שורה או עמודה או אלכסון. מצב תיקו הוא מצב בו אף שחקן לא הצליח לייצר מצב של ניצחון ולא נותר עוד תא פנוי בלוח המשחק. במימוש הנוכחי, מצב ניצחון בשורה באלכסון הראשי, לדוגמא, יתרחש במידה וערכי האיברים הראשון, החמישי והתשיעי ברשימה זהים (נסו להבין מדוע).

מהו מצב תיקו?

מצב תיקו הוא מצב בו אף שחקן לא הצליח לייצר מצב של ניצחון ולא נותר עוד תא פנוי בלוח המשחק. במימוש הנוכחי, מצב תיקו הינו מצב בו אין ניצחון לאף שחקן וכל האיברים ברשימה שונים מ-<sup>8</sup>#.

## פתרון מוצע<sup>9</sup>

בדיקת תנאי ניצחון, עבור האלכסון הראשי, למשל, ייעשה באופן הבא:



בדיקת מצב משחק (ניצחון/תיקו/המשך משחק):

<sup>8</sup> כזכור הסימן '#' מייצג תא ריק  
<sup>9</sup> הפתרון המוצע כאן הינו חלקי. נדרש לקרוא לפעולה עבור כל אחת מהתאים בלוח.

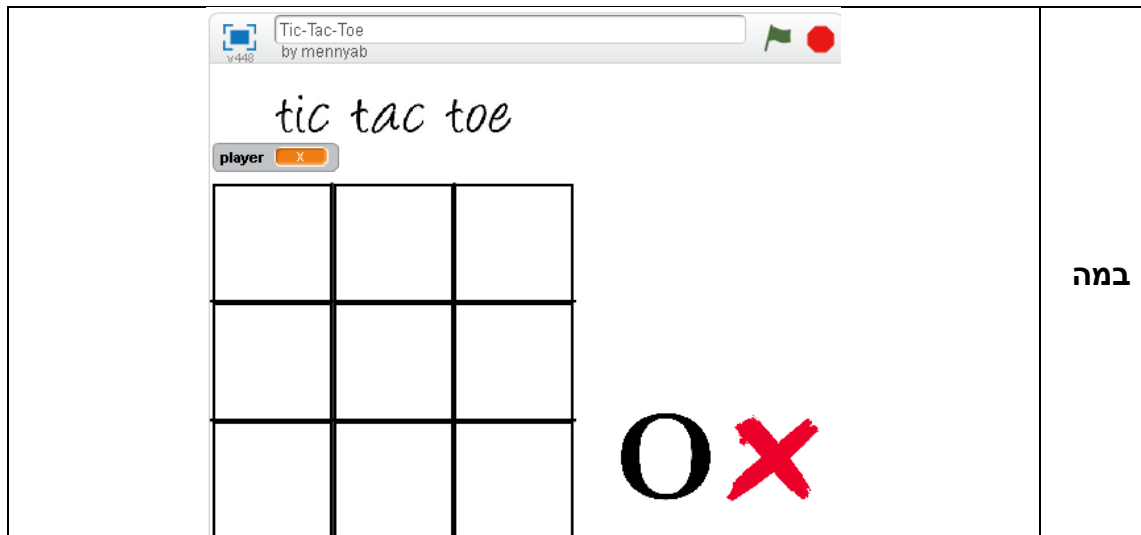
בדיקת תנאי  
ניצחון

```
define CheckWinner player
  then
    if player = X then
      broadcast playerO_wins
    else
      broadcast playerX_wins
  else
    if board contains value ? then
      broadcast keep_on_playing
    else
      broadcast tie
```

This method checks if the player wins. Otherwise, returns the game state: continue playing or tie.

1. ניהול מתואם של מסרים מתא מסויים לשחקן ומשחקן ללוח המשחק.
2. אתגר: ניהול טורניר בין השחקנים. למשל, הטוב מ-5 משחקים והכרזת המנצח.
3. אתגר: הפיכת מבנה המשחק להיות מנוהל באופן סינכרוני (באמצעות משתני ענן).

### הצעת פתרון





The image displays a vertical stack of Scratch code blocks on a light gray grid background. The code is organized into two columns. The left column starts with a 'when clicked' event block, followed by a 'go to x: 271 y: -123' block. The right column starts with a 'when I receive show\_playerX\_c1' event block, followed by a 'create clone of myself' block and a 'go to x: -111 y: 45' block. Below these are nine identical blocks, each triggered by a 'when I receive' event with a unique ID (show\_playerX\_c2 through show\_playerX\_c9). Each block contains a 'create clone of myself' block and a 'go to x: [value] y: [value]' block with specific coordinates.

Event	Action	X	Y
when clicked	go to x: 271 y: -123	271	-123
when I receive show_playerX_c1	create clone of myself	-111	45
when I receive show_playerX_c2	create clone of myself	-18	44
when I receive show_playerX_c3	create clone of myself	74	44
when I receive show_playerX_c4	create clone of myself	-112	-47
when I receive show_playerX_c5	create clone of myself	-19	-46
when I receive show_playerX_c6	create clone of myself	72	-45
when I receive show_playerX_c7	create clone of myself	-110	-137
when I receive show_playerX_c8	create clone of myself	-19	-135
when I receive show_playerX_c9	create clone of myself	74	-136

X

The image displays a Scratch script editor with a sequence of nine event-driven code blocks. Each event block is followed by a 'create clone of myself' block and a 'go to x: y:' block. The events are: 'when clicked', 'when I receive show\_playerO\_c1', 'when I receive show\_playerO\_c2', 'when I receive show\_playerO\_c3', 'when I receive show\_playerO\_c4', 'when I receive show\_playerO\_c5', 'when I receive show\_playerO\_c6', 'when I receive show\_playerO\_c7', 'when I receive show\_playerO\_c8', and 'when I receive show\_playerO\_c9'. The 'go to x: y:' blocks specify coordinates for each clone.

```
when clicked
  go to x: 20 y: -182

when I receive show_playerO_c1
  create clone of myself
  go to x: -282 y: -15

when I receive show_playerO_c2
  create clone of myself
  go to x: -187 y: -15

when I receive show_playerO_c3
  create clone of myself
  go to x: -95 y: -18

when I receive show_playerO_c4
  create clone of myself
  go to x: -280 y: -107

when I receive show_playerO_c5
  create clone of myself
  go to x: -186 y: -106

when I receive show_playerO_c6
  create clone of myself
  go to x: -96 y: -108

when I receive show_playerO_c7
  create clone of myself
  go to x: -279 y: -197

when I receive show_playerO_c8
  create clone of myself
  go to x: -188 y: -197

when I receive show_playerO_c9
  create clone of myself
  go to x: -95 y: -198
```

O

The image shows a Scratch script with two columns of code blocks. The left column contains four 'when I receive' blocks: 'when clicked', 'when I receive keep\_on\_playing', 'when I receive x\_to\_start', and 'when I receive o\_to\_start'. Each block is followed by 'switch costume to hide', a 'say' block with a 2-second duration, and a 'hide' block. The right column contains four 'when I receive' blocks: 'when I receive playerO\_wins', 'when I receive playerX\_wins', and 'when I receive tie'. Each of these blocks is followed by 'switch costume to text awesome', a 'go to x: 159 y: 133' block, a 'set size to 100 %' block, a 'show' block, a 'repeat 10' loop containing a 'change size by 5' block, and a 'say' block with a 5-second duration. The 'tie' block's 'say' block contains the text 'It's a tie. Try again'.

```
when clicked
  go to x: 176 y: -27
  show

when I receive keep_on_playing
  say It's a tie. Keep on going! for 2 secs

when I receive x_to_start
  switch costume to hide
  say Player X starts for 2 secs
  hide

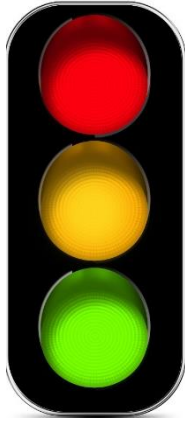
when I receive o_to_start
  switch costume to hide
  say Player O starts for 2 secs
  hide

when I receive playerO_wins
  switch costume to text awesome
  go to x: 159 y: 133
  set size to 100 %
  show
  repeat 10
    change size by 5
  say Player O wins! for 5 secs

when I receive playerX_wins
  switch costume to text awesome
  go to x: 159 y: 133
  set size to 100 %
  show
  repeat 10
    change size by 5
  say Player X wins! for 5 secs

when I receive tie
  switch costume to text awesome
  go to x: 159 y: 133
  set size to 100 %
  show
  repeat 10
    change size by 5
  say It's a tie. Try again for 5 secs
```

test



## משימת רמזור (Traffic Light)

יש ליצור באמצעות סביבת scratch עמוד רמזור.

עמוד הרמזור יורכב ממלבן ובו 3 עיגולים, אשר מדמה את פעולת הרמזור:

אדום, כתום וירוק.

דמות הרמזור תנהג באופן המעגלי כך: 30 שניות נורה אדומה קבועה, שנייה

אחת נורה כתומה ובמשך 15 שניות נורה ירוקה קבועה.

בסיום פעולת הרמזור, הנורה הירוקה תהבהב במשך 5 שניות, לאחר מכן, הנורה הכתומה

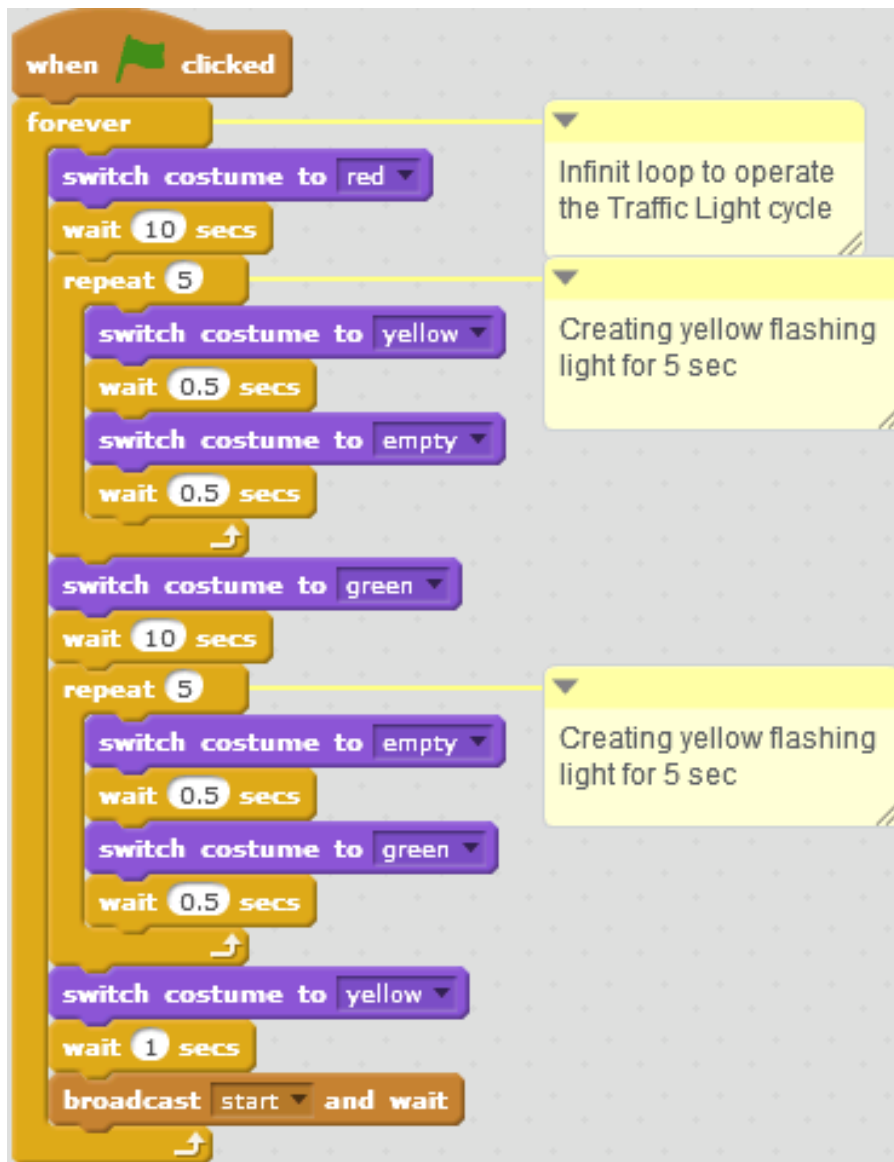
תוצג במשך שנייה אחת ואז הנורה תתחלף לאדומה.

### הערות:

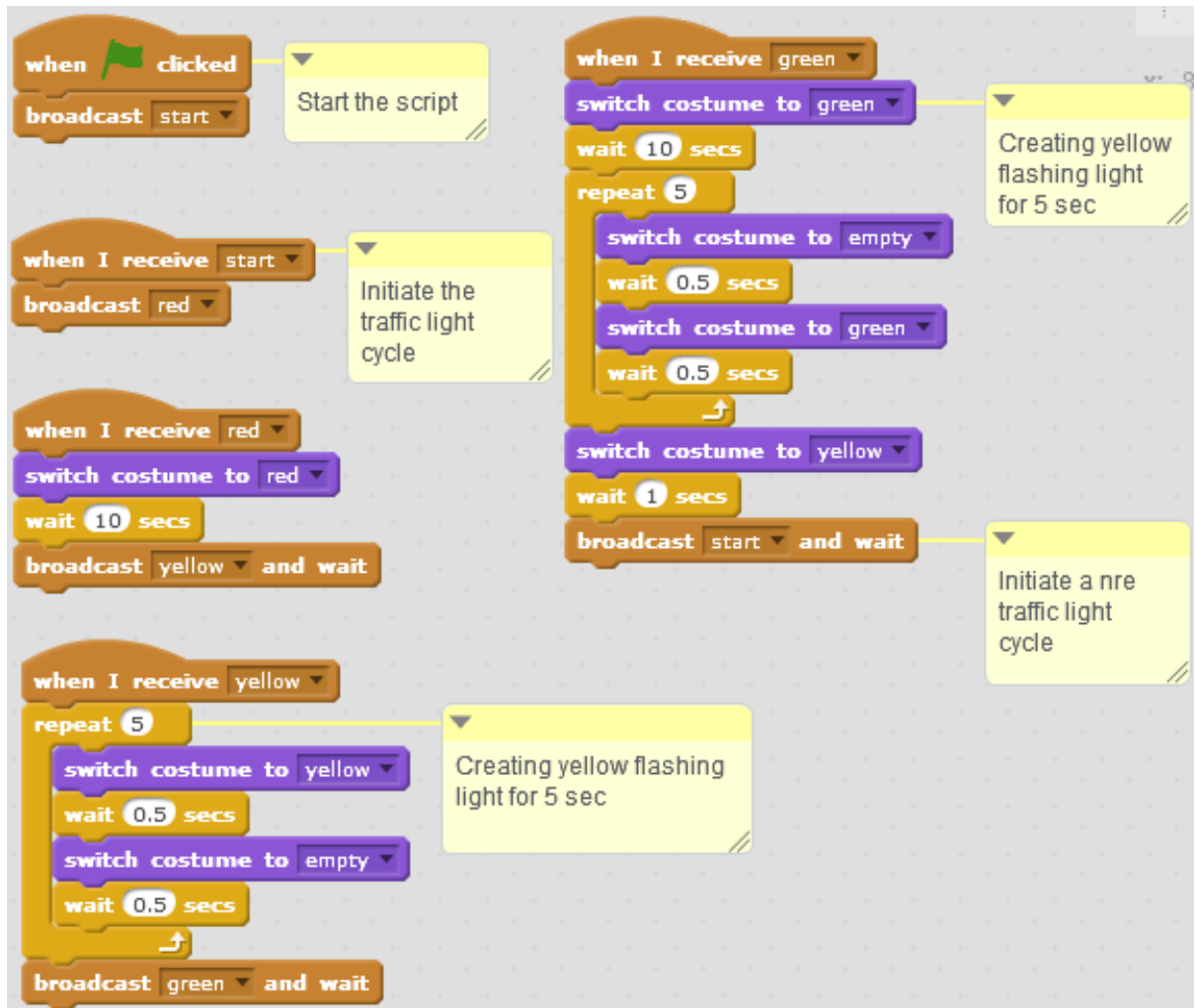
- משך זמן פעילות מומלץ וניתן להגדיר מחזור זמנים שונה.
- מומלץ להשתמש במנגנון שינוי costume (תלבושת) הנמצא בתפריט Looks (בצבע סגול)

בסיום המטלה, יש לשתף את הפתרון בסטודיו:

<https://scratch.mit.edu/studios/2835403/>



<sup>11</sup> הפתרון מבוסס על מנגנון ביצוע חוזר של החלפת תלבושות (costume)



<sup>12</sup> הפתרון מבוסס על מנגנון מסרים (messaging)