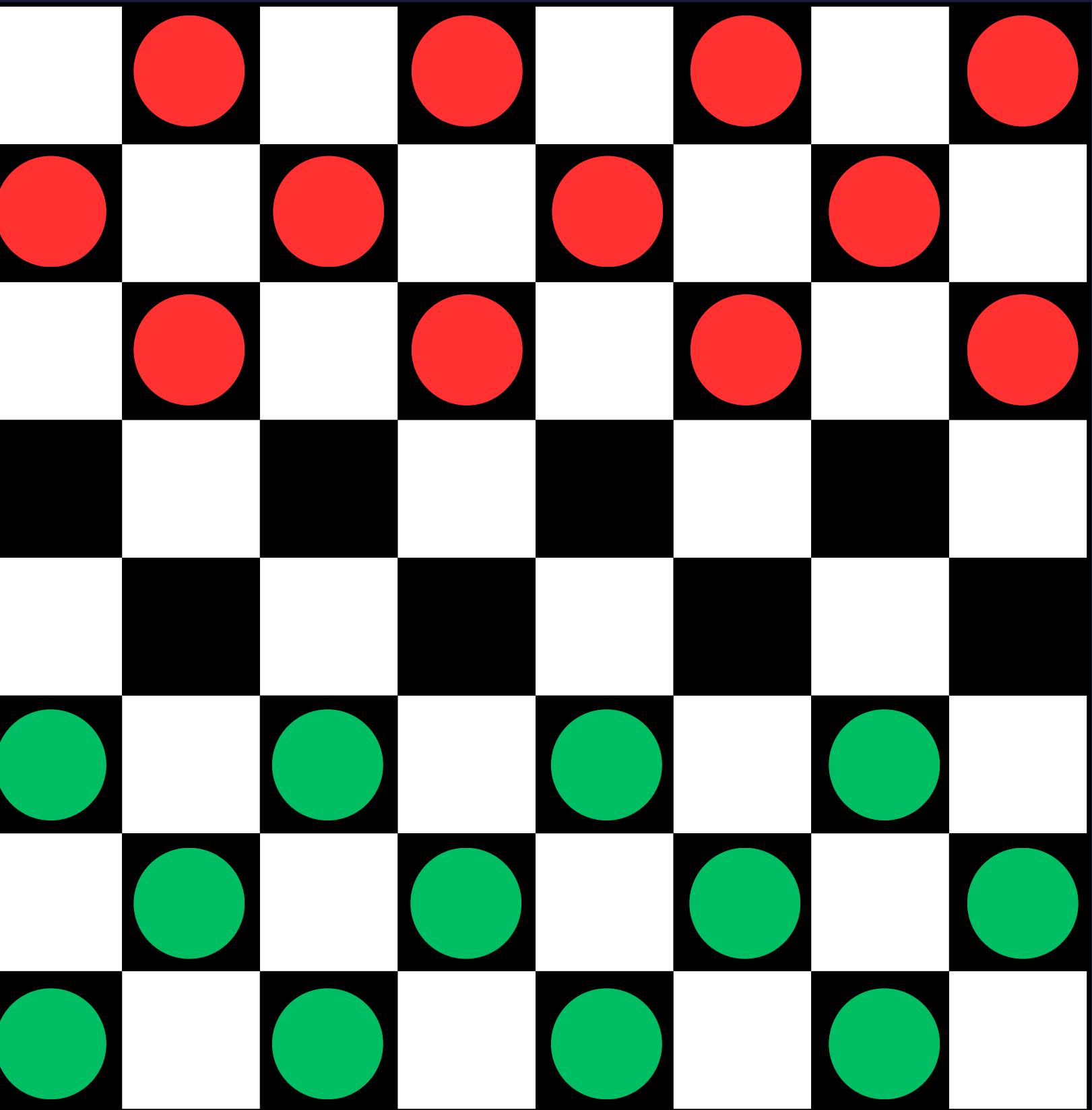


# Enhanced Checkers

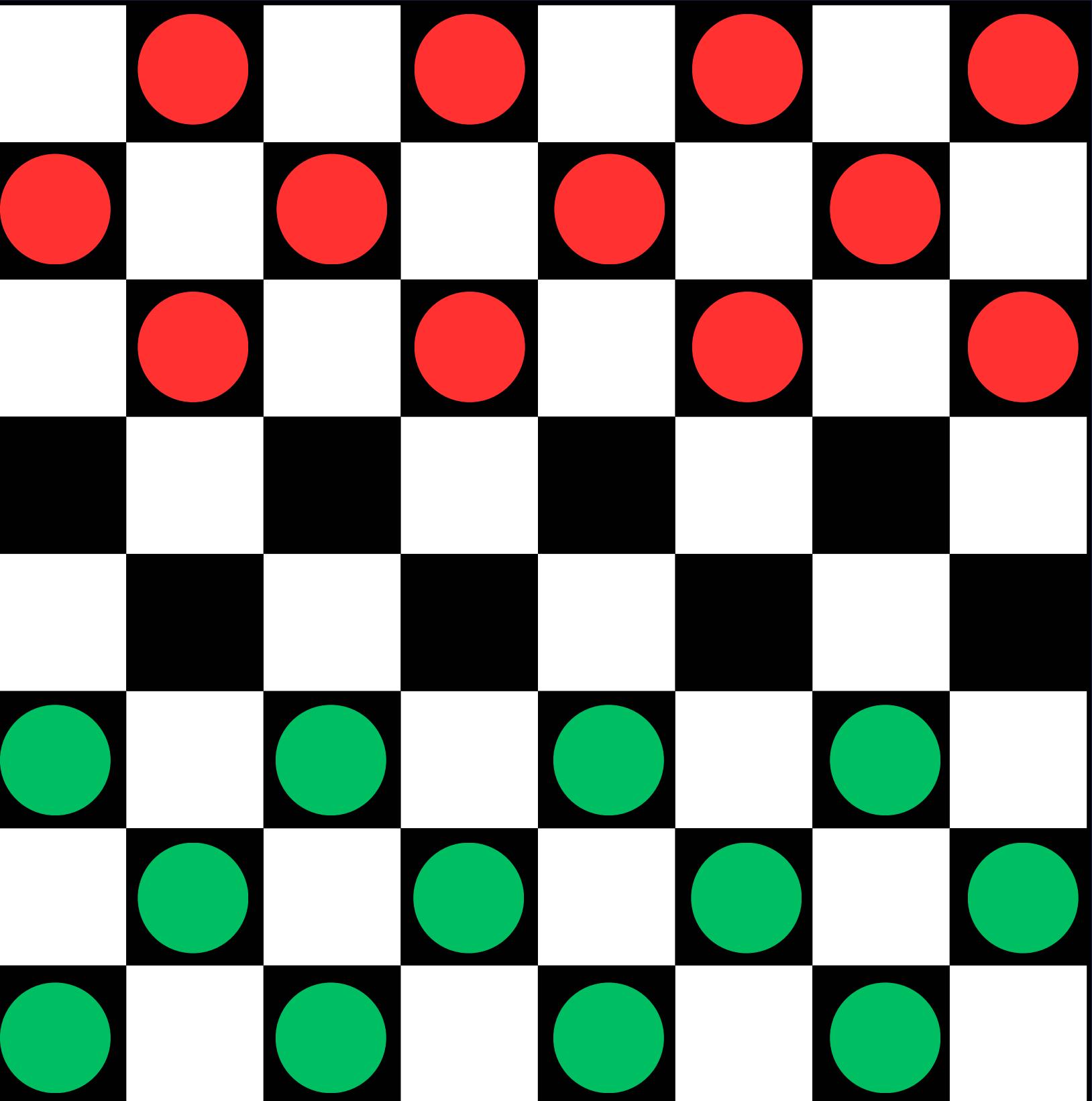
אלגוריתמים מתקדמים לתוכנו  
ותזמון מערכות נבונות



# Enhanced Checkers

עלآل יחיא  
מוחמד אבו פול  
מוחמד לוחאמ

מרצה ד"ר רינה צביאל גירשין

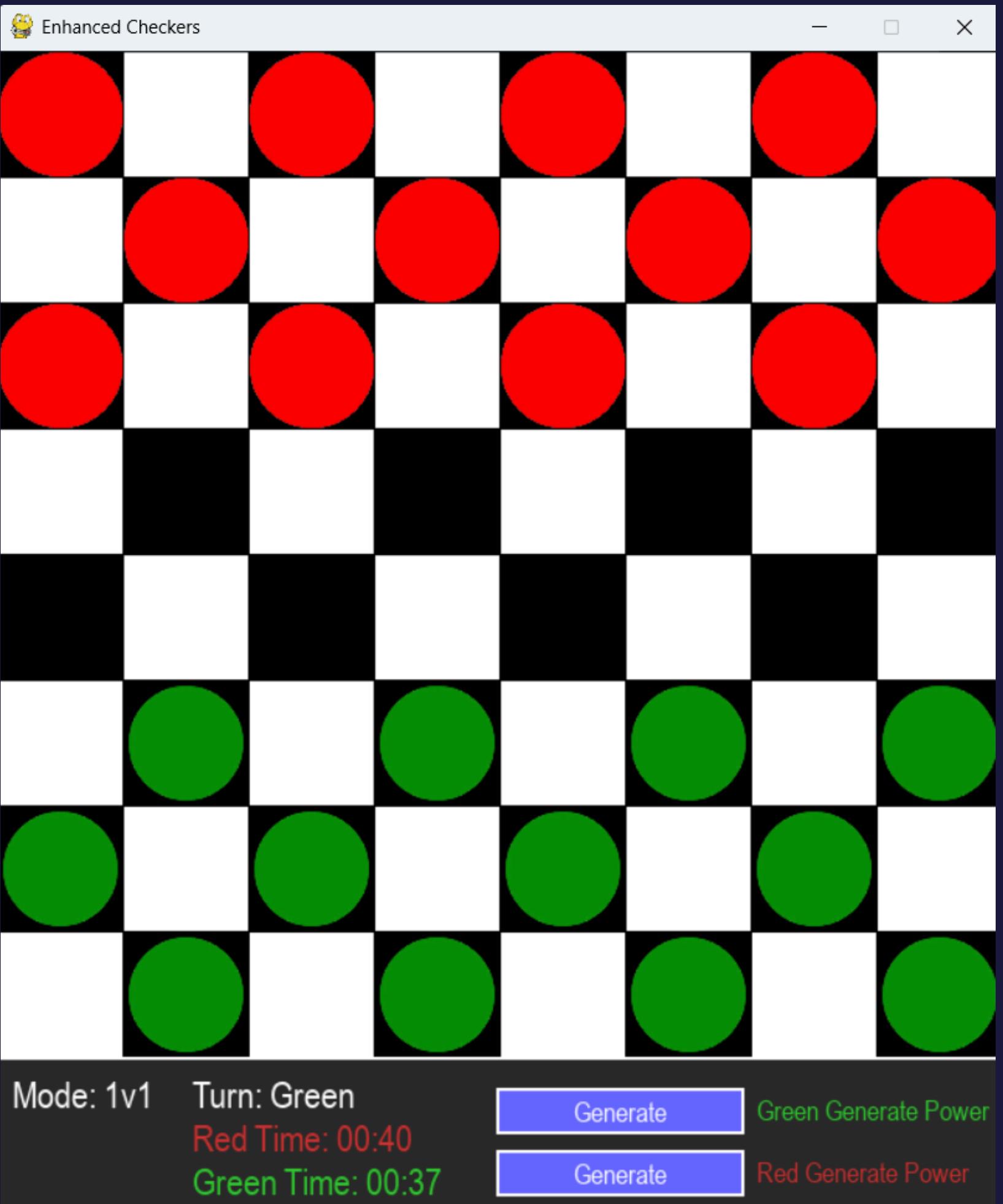


# מה מישנו

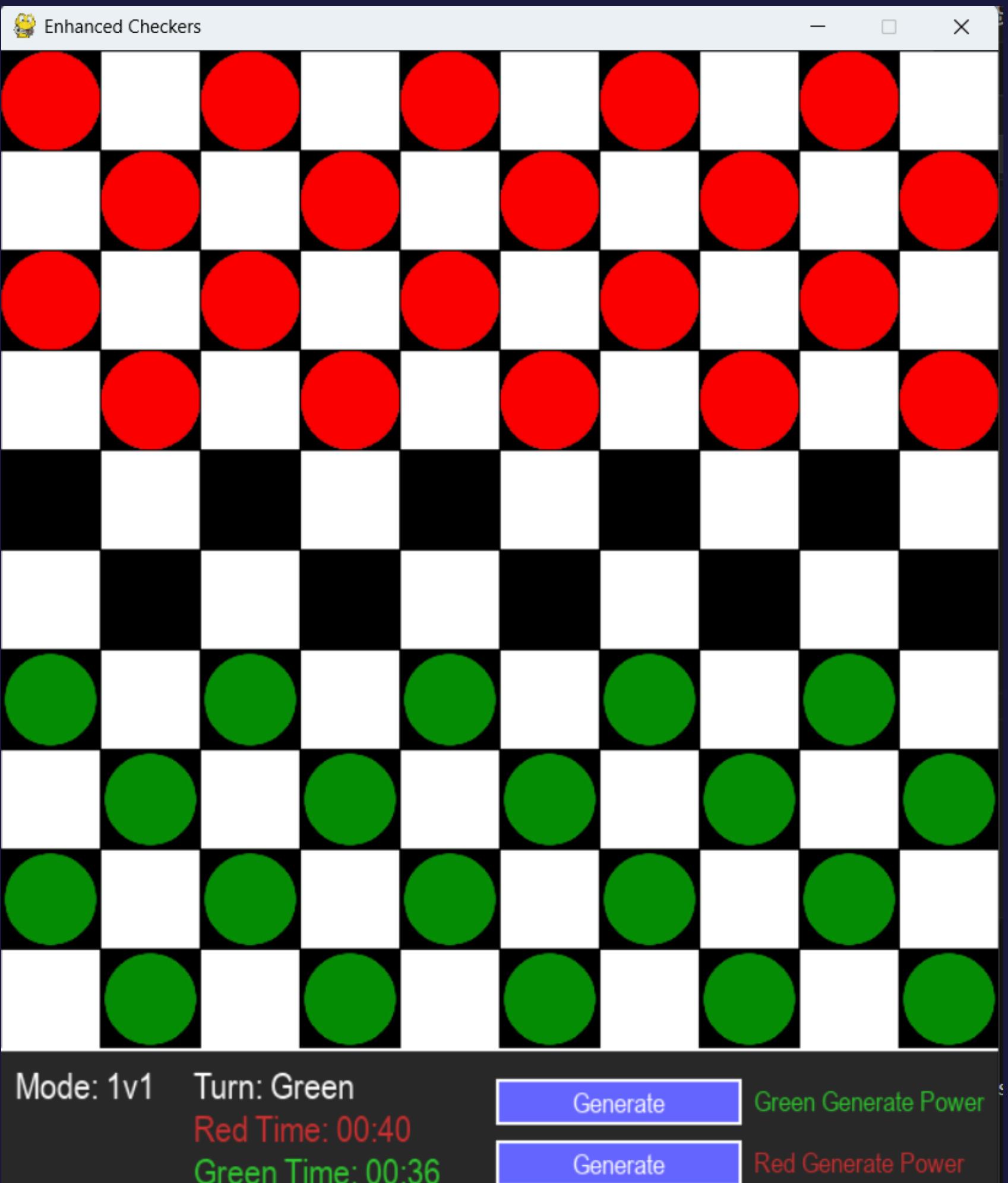
- משחק גרפי מלא ב-Pygame
- לוח דינמי:  $12 \times 12$ ,  $10 \times 10$ ,  $8 \times 8$
- שני מצבים: PvAI ו-PvP
- מערכת כוחות מיוחדים (MOVE\_TWICE)  DOUBLE\_JUMP , IMMUNITY 
- MOVE\_TWICE (MOVE\_TWICE, FORCE\_SWAP)  
- בינה מלאכותית המבוססת על Alpha-Beta- Minimax

# תחילת המשחק

- אפשרות לבחור גודל הלוח של המשחק (8x8, 10x10, 12x12)
- אפשרות לבחור לשחק נגד שחקן רגיל (PvP), או נגד בינה מלאכותית (PvAI)
- אם נבחר בנגד AI, יש אפשרות לבחור אלגוריתם שישתמש בו ה-AI (אלגוריתם Alpha-Beta או Minimax)
- אחרי בחרת האלגוריתם יש אפשרות לבחור העומק שישתכל אליו ה-AI מ-1 עד 8

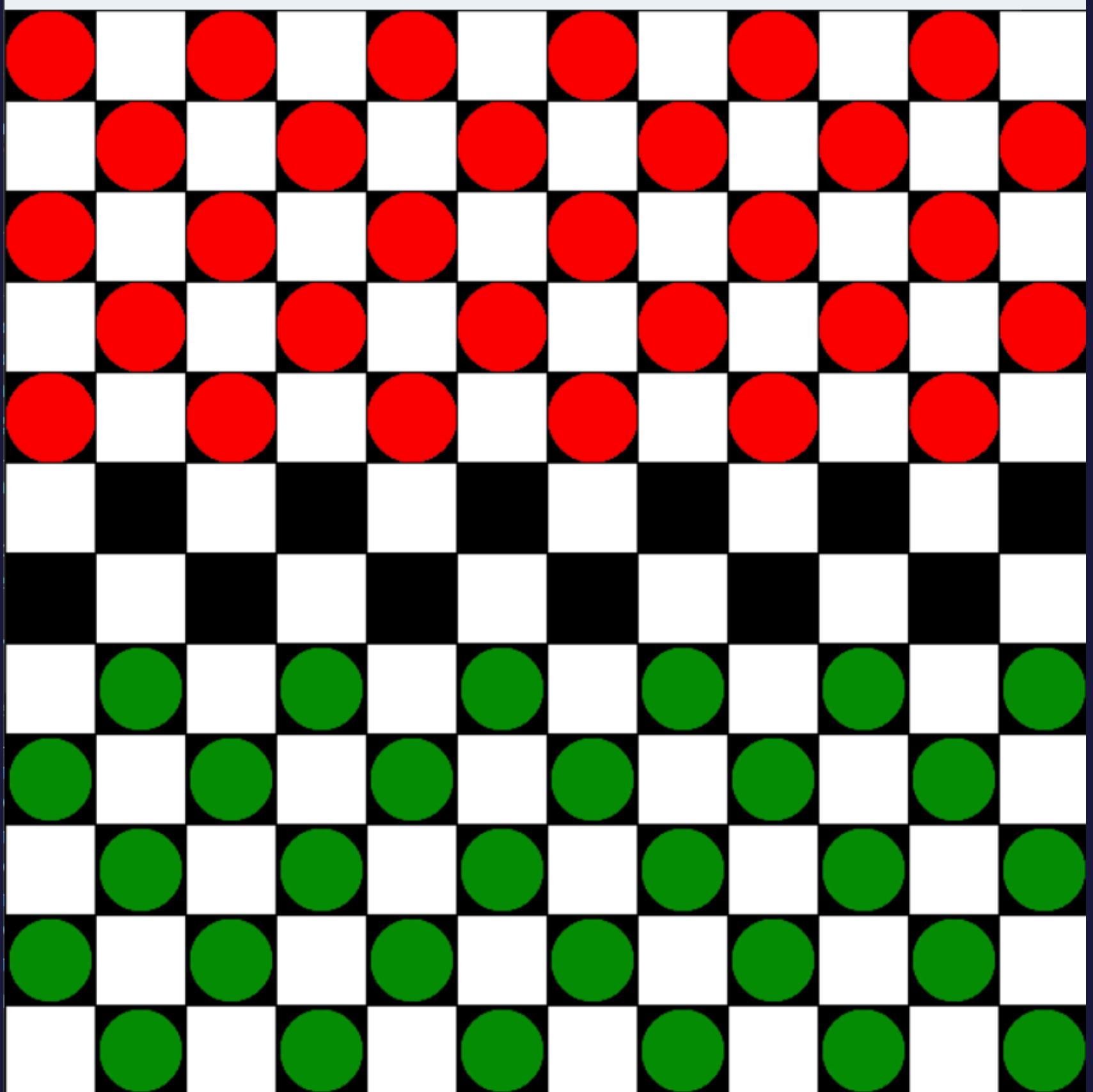


# גודל הלוח 8x8



גודל הלוח

 Enhanced Checkers



Mode: 1v1

Turn: Green

Red Time: 00:40

Green Time: 00:36

Generate

Green Generate Power

Generate

Red Generate Power

גודל הלוח

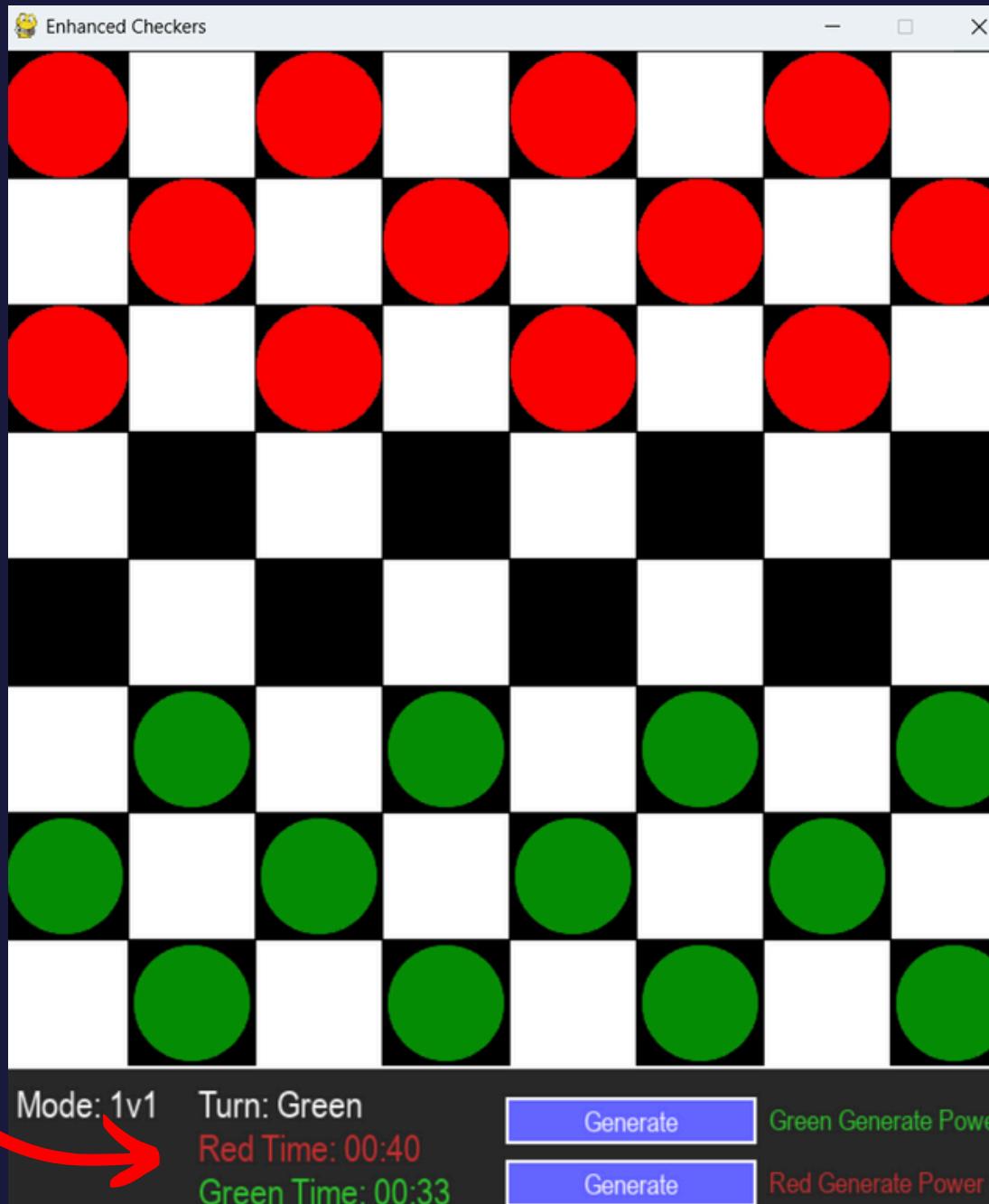
# כללי המשחק

- שני שחקנים (אדום מלמעלה, ירוק מלמטה), P v P או AI v P
- תור בן 40 שניות
- תנועה באלבוסון, אכילה מקפיצה
- אכילה לאחורה גם לשחקן רגיל (לא רק למלך)
- קידום למלך (הגעה לבסיס היריב)
- קידום למלך-על (3 אכילות רצופות)
- שימוש בכוחות מיוחדים (כל שחקן יכול להגריל כוח רנדומלי פעם אחת במהלך המשחק ולהשתמש בו בתור הנוכחי שלו)
- ניצחון: חיסול כל כלי היריב או חסימת תנועות

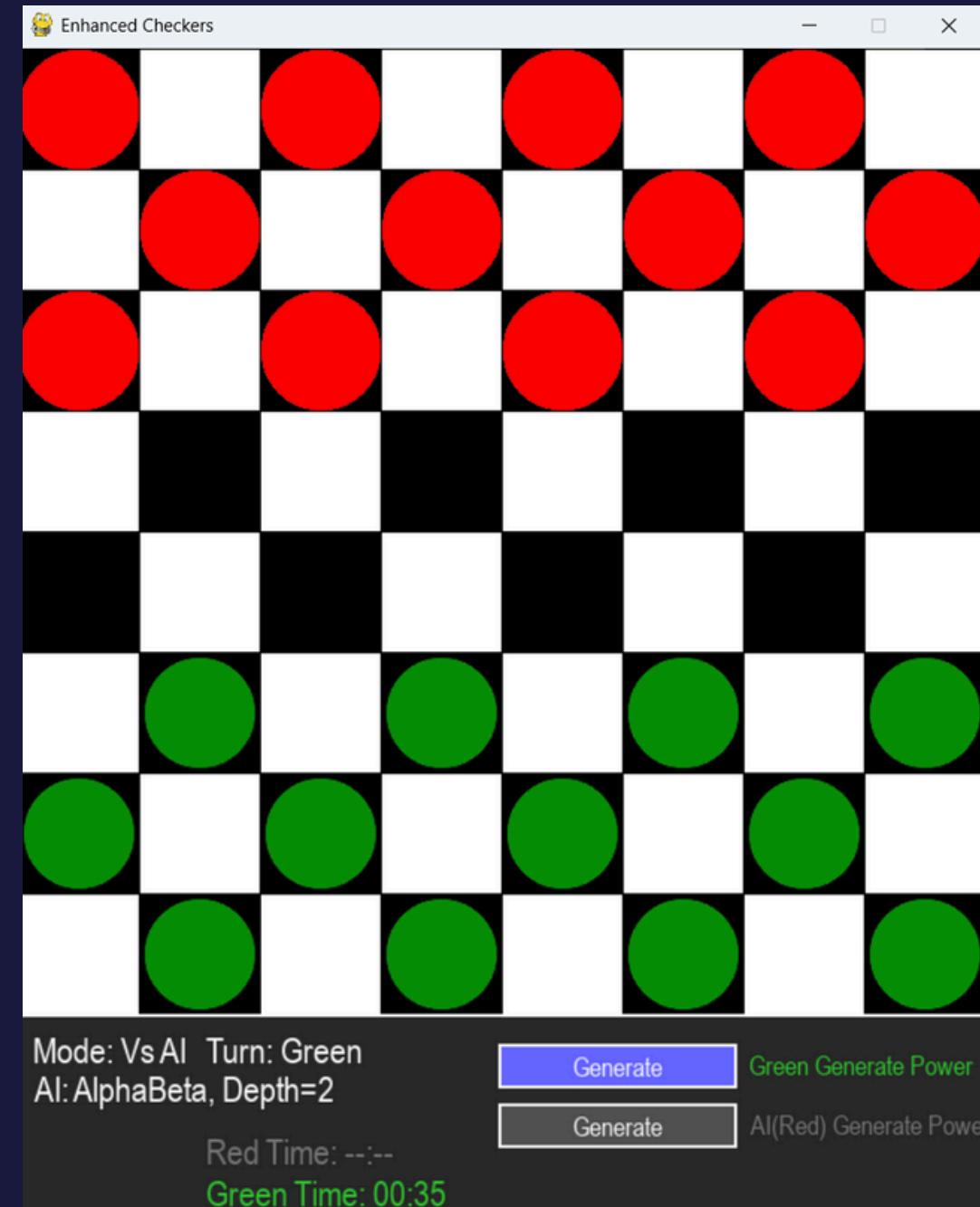
# כללי המשחק

- שבי שחנים (אדום מלמעלה, ירוק מלמטה), ירוק מלמטה)

PvP



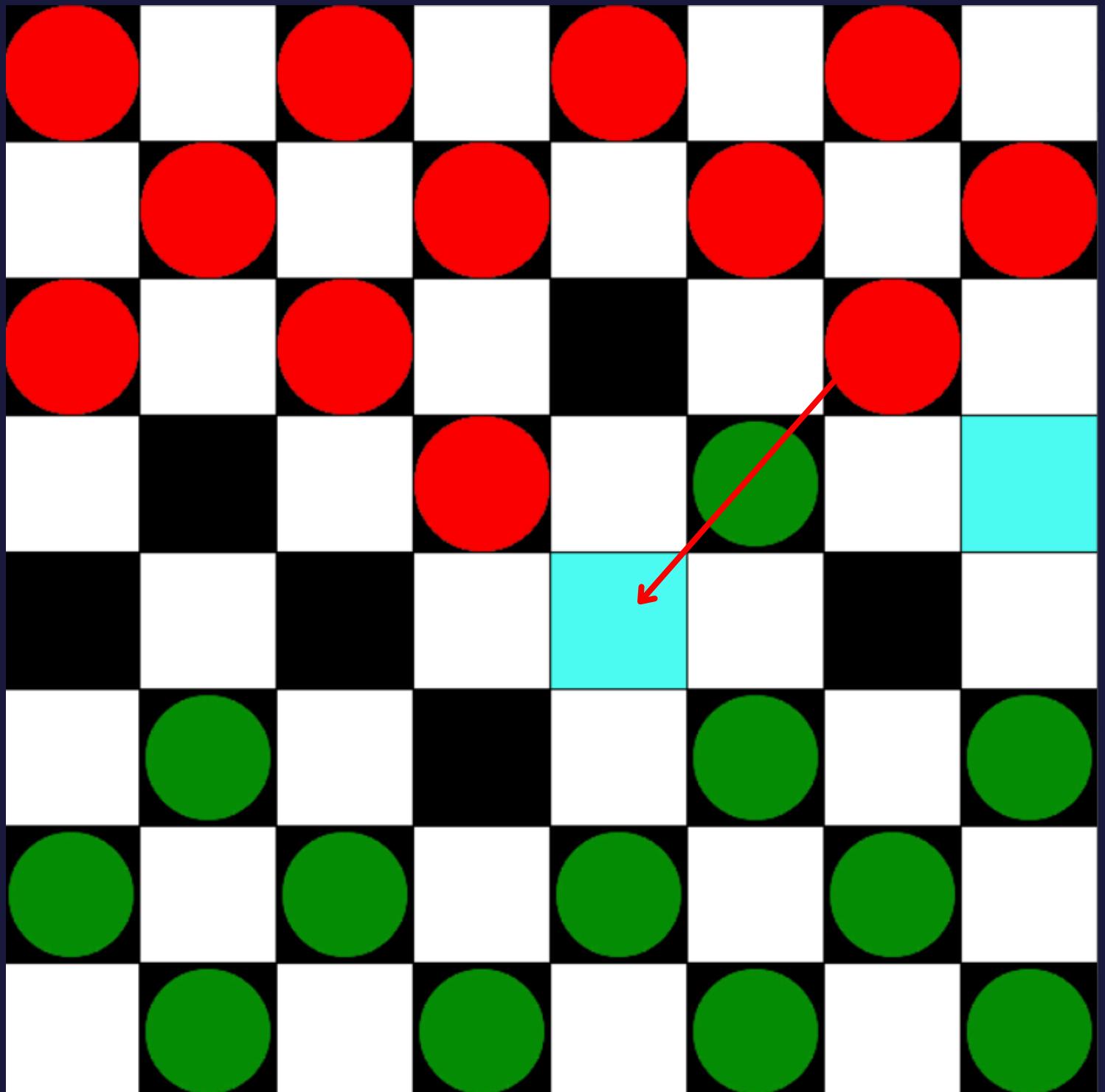
PvAI



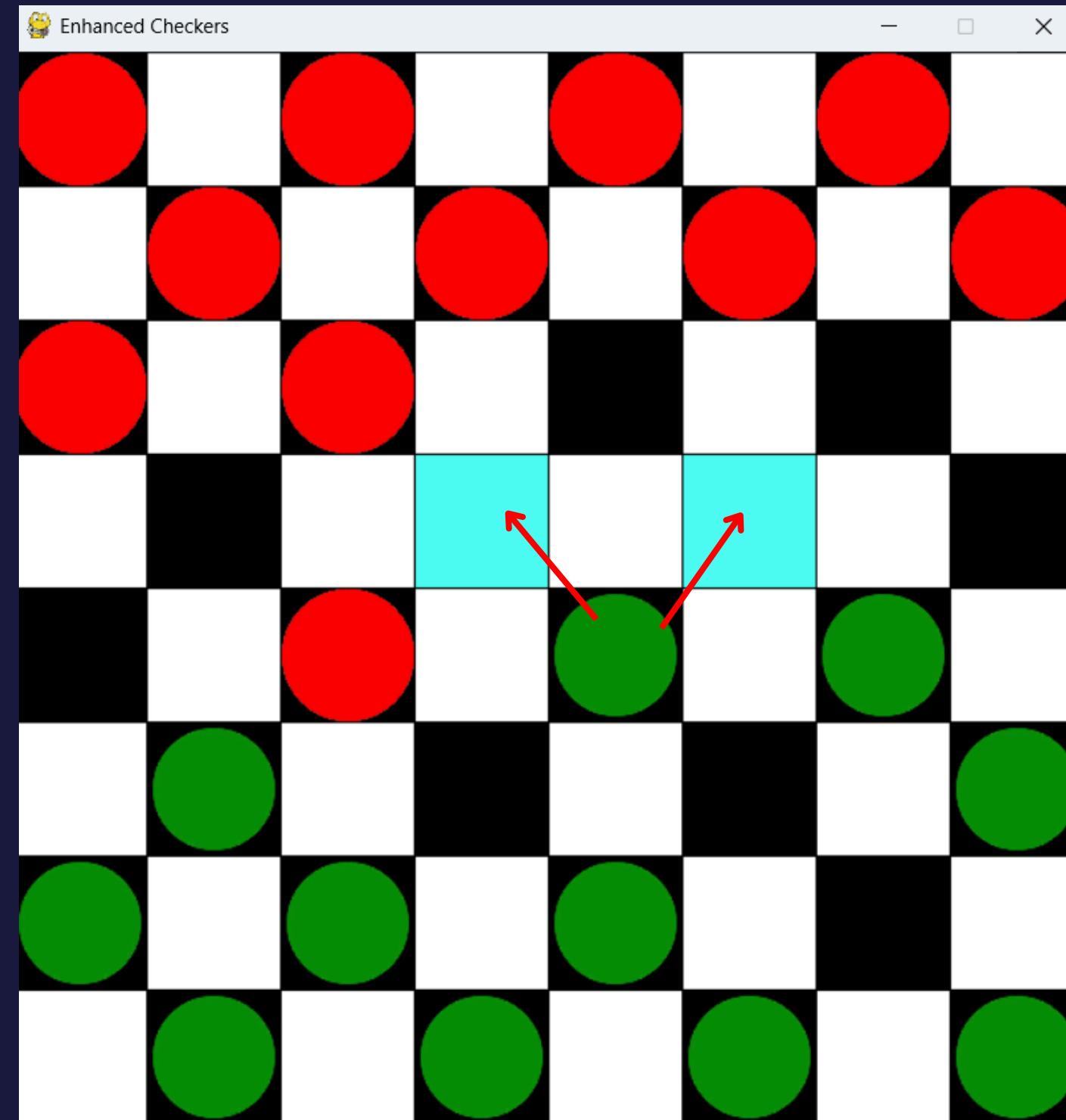
טור בן 40 שניות

# כללי המשחק

- **אכילה מזקיפה**

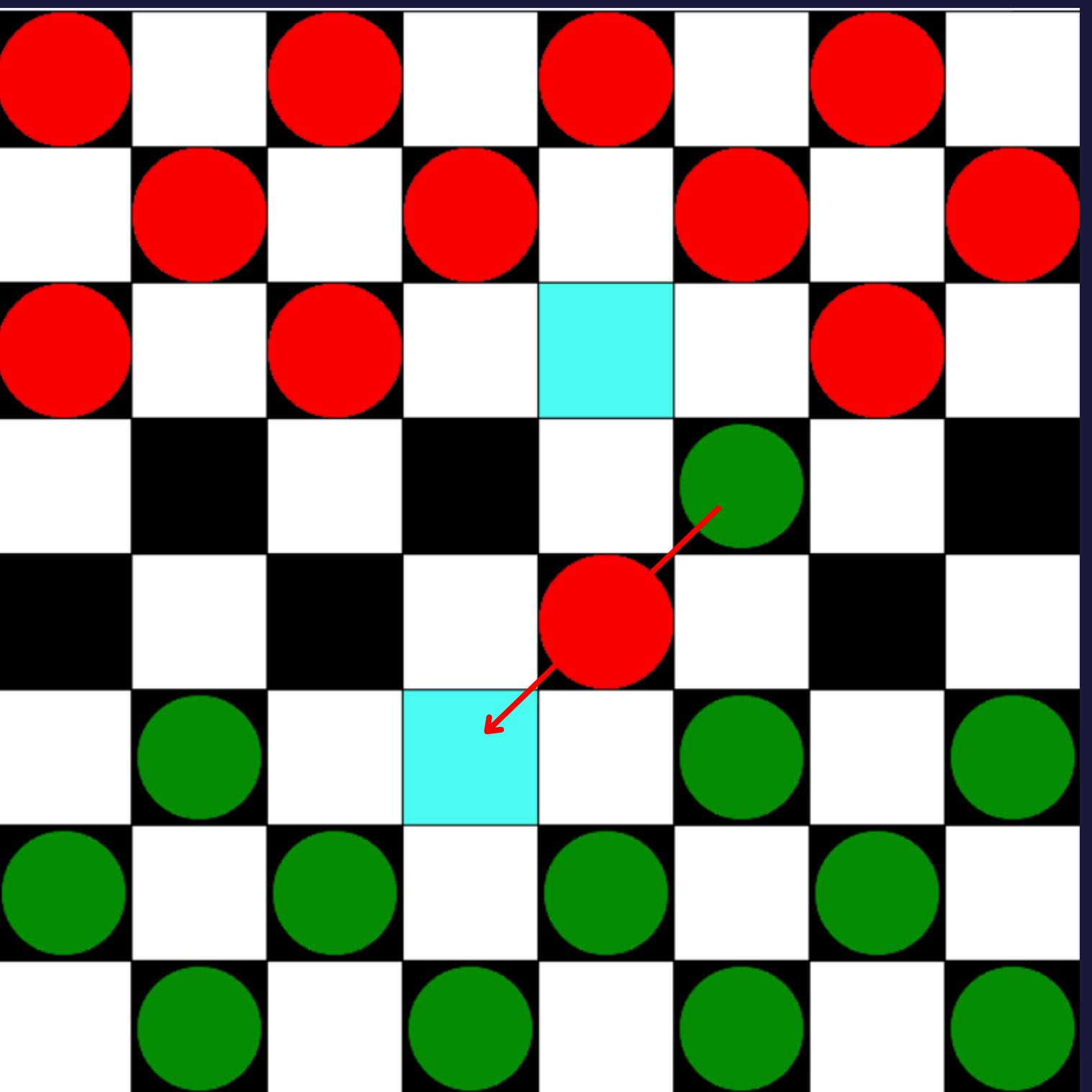


- **תנועה באלכסון**



# כללי המשחק

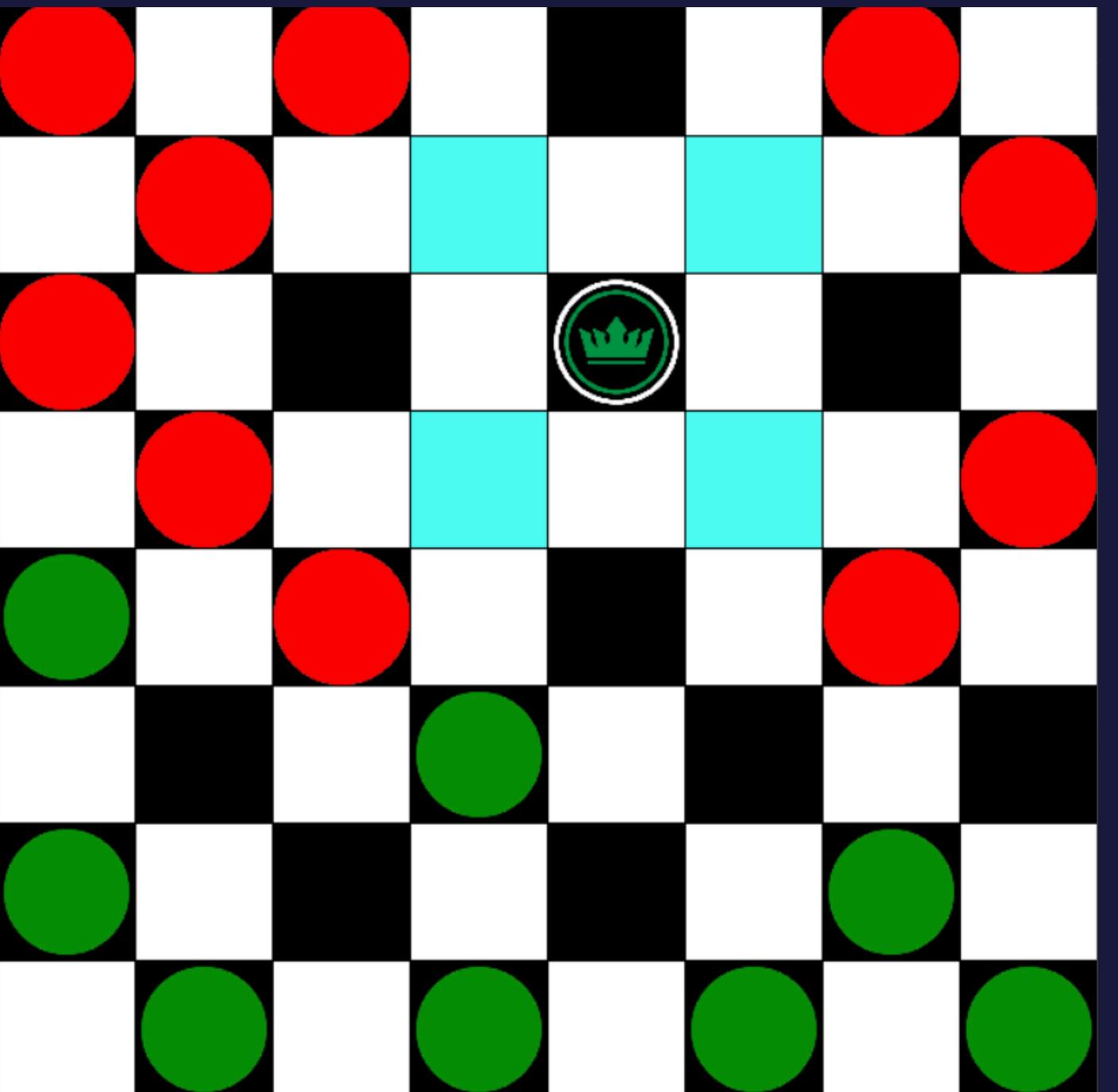
- אכילה לאחורה גם לשחקן רגיל (לא רק למלר)



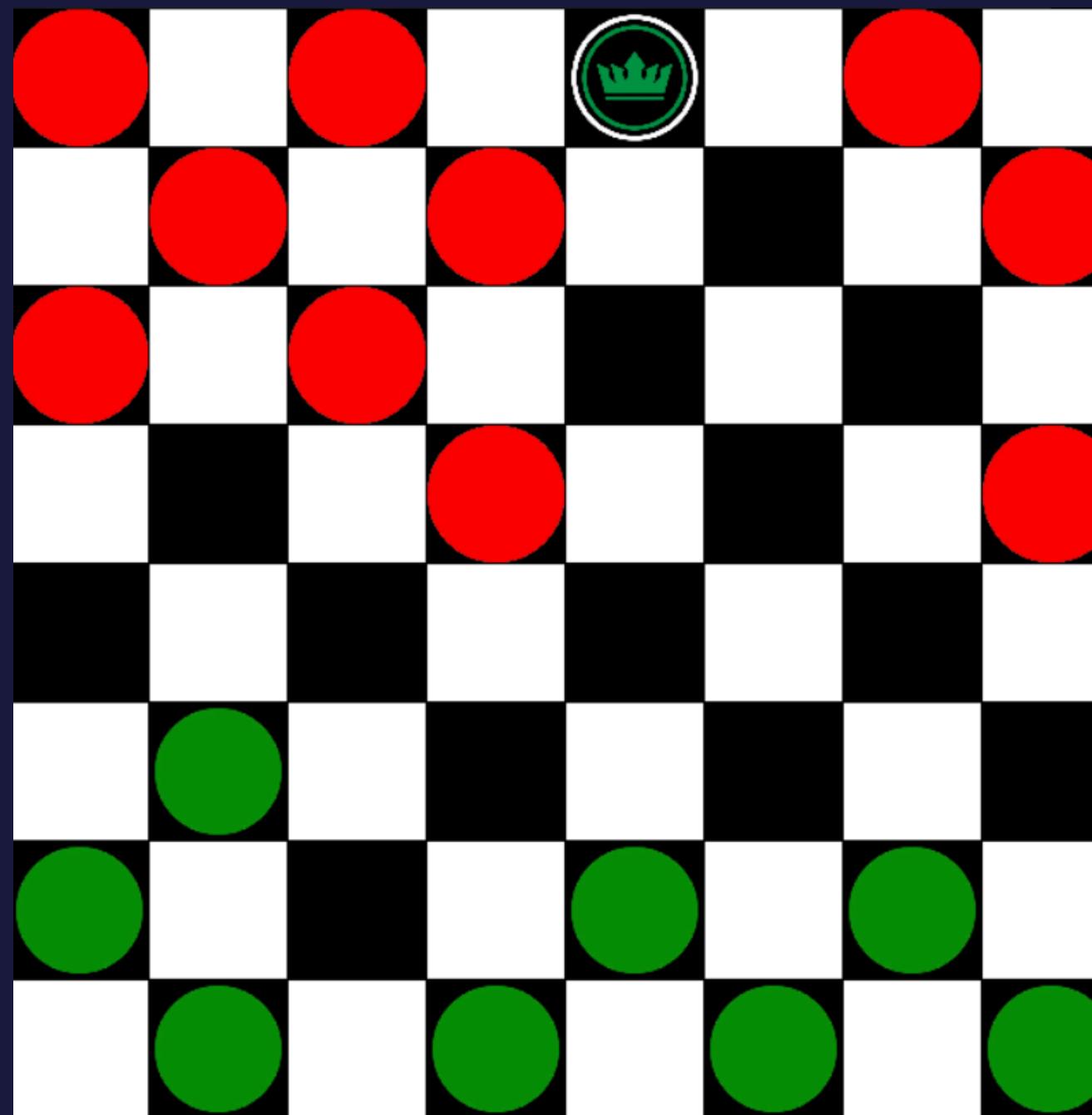
משנה את  
פונקציית  
הערכתה

# כללי המשחק

## תנועת המלך

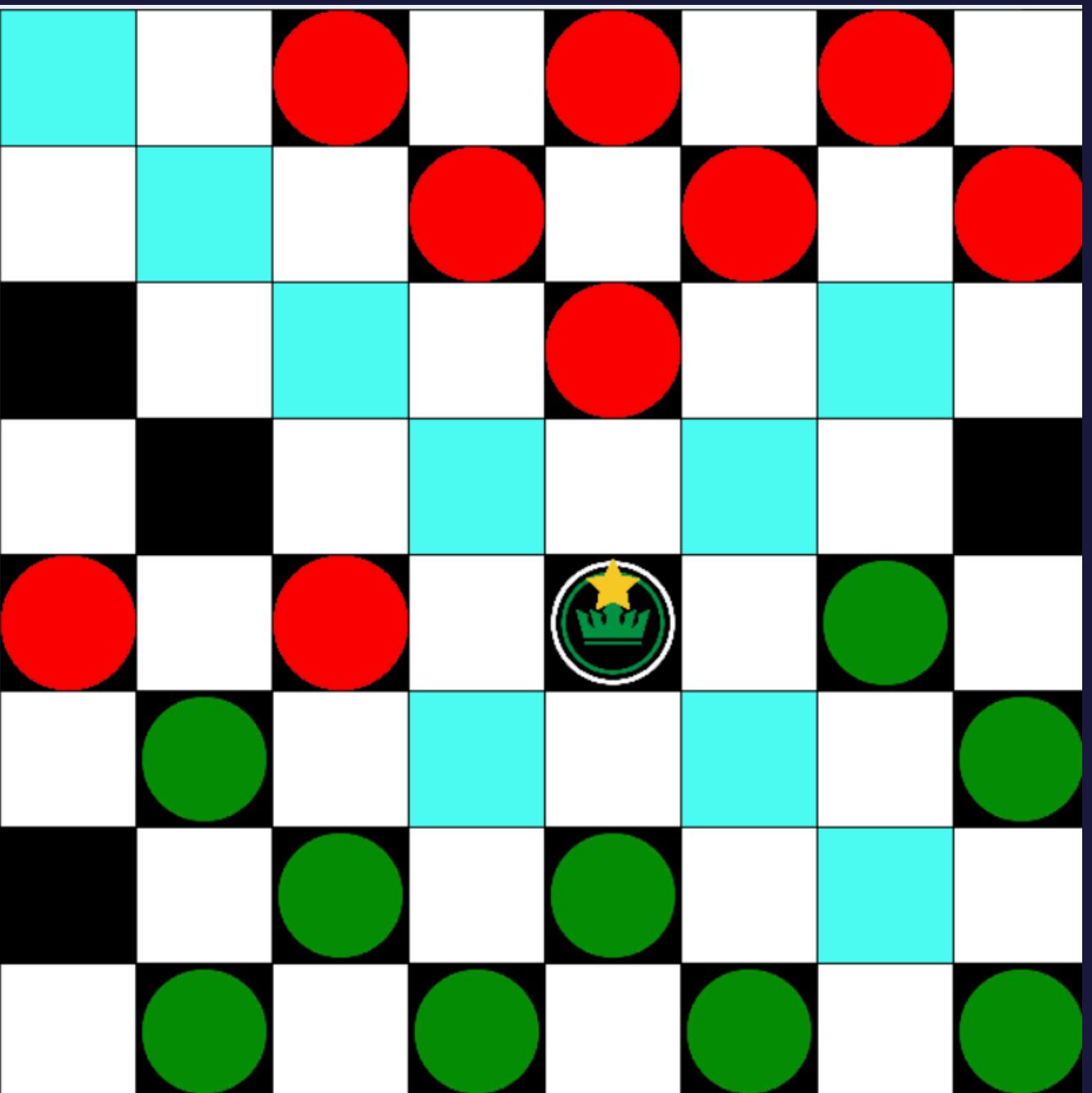


- קידום מלך (העה לביסיס היריב)

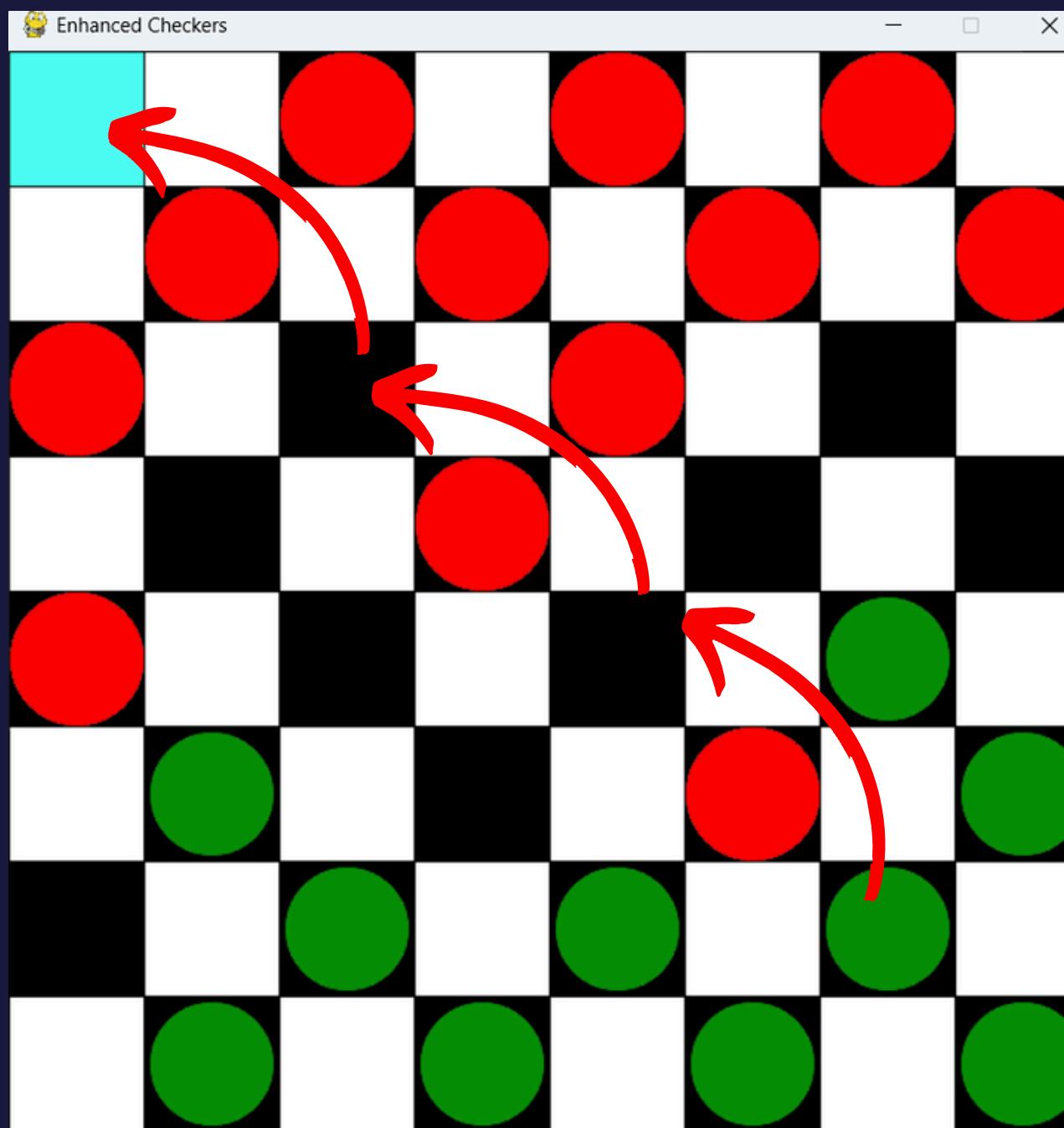


# כללי המשחק

## תנועת המלך על



- קידום למלך-על (3 אכילות רצופות)

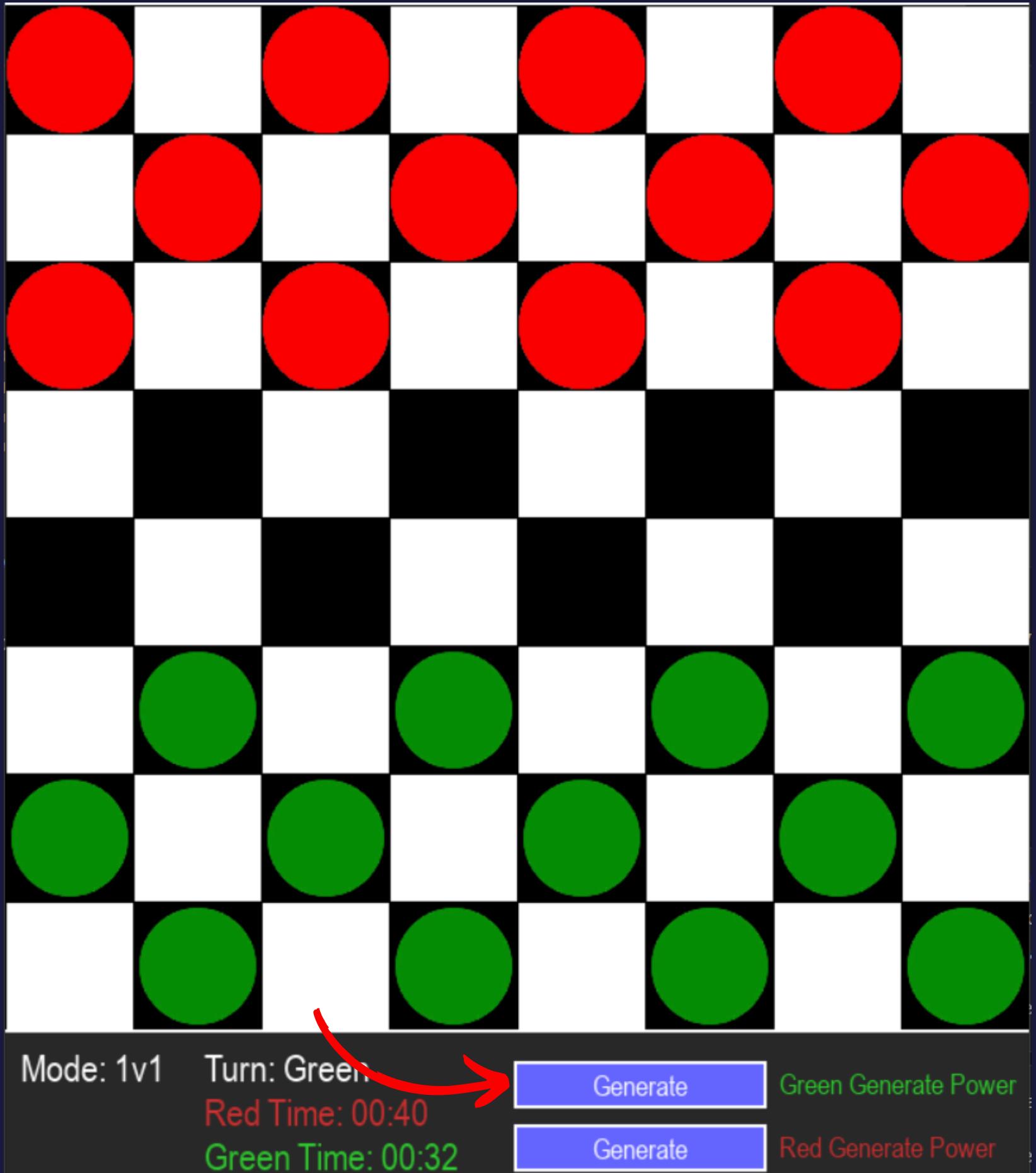


# כללי המשחק

- שימוש בכוחות מיוחדים:

בלחיצה על כפתור Generate מתבצעת בחירת כוח רנדומלית לכל שחקן פעם אחת במהלך המשחק.

הכוח מופיע עם הסבר איך להשתמש בו אחרי הלחיצה על Generate.



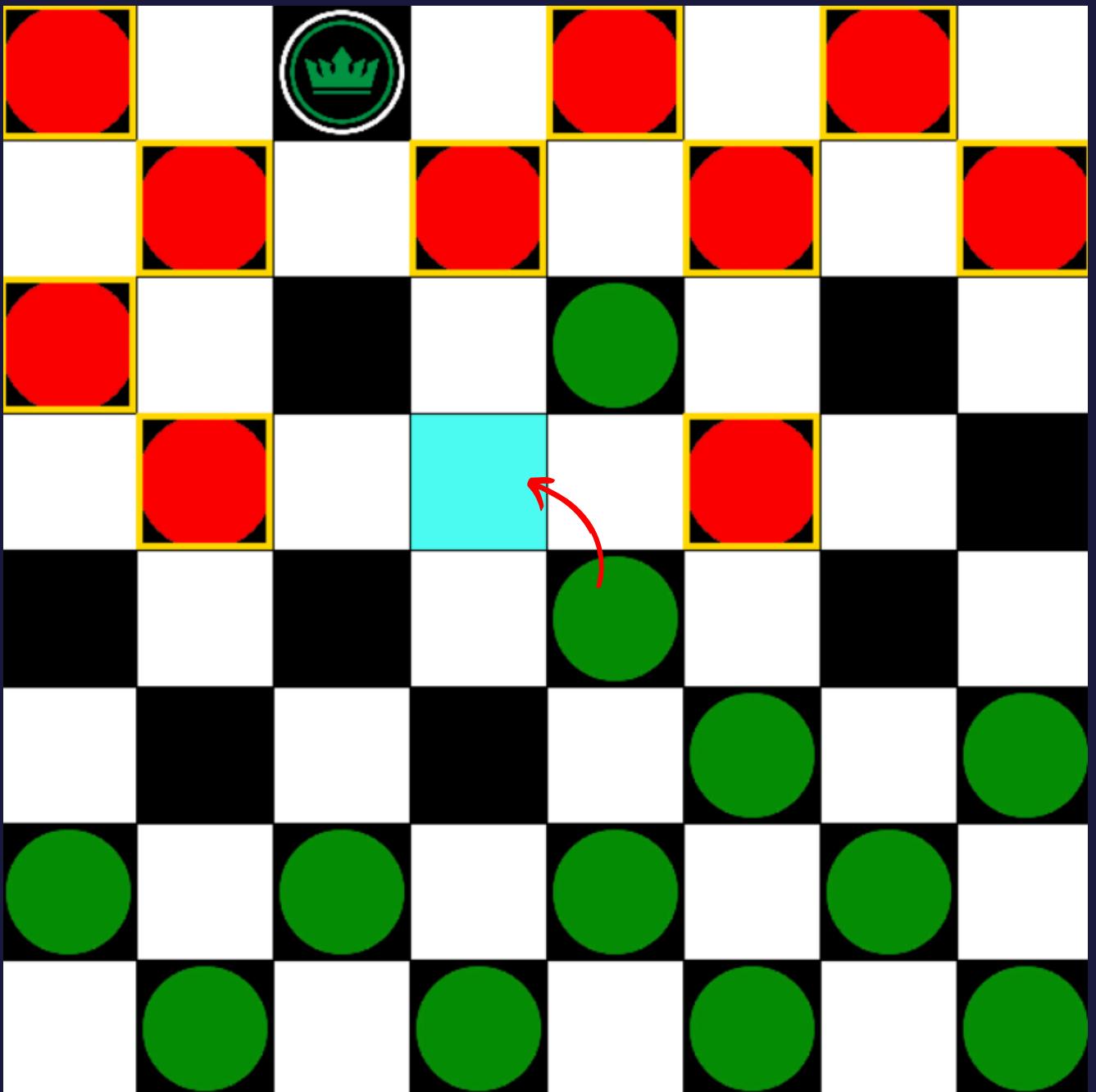
# כללי המשחק



• שימוש בכוחות מיוחדים: IMMUNITY:

**שימוש בכוח זה מאפשר לשחקנים שלהם הגנה על השחקנים שלו** בתור

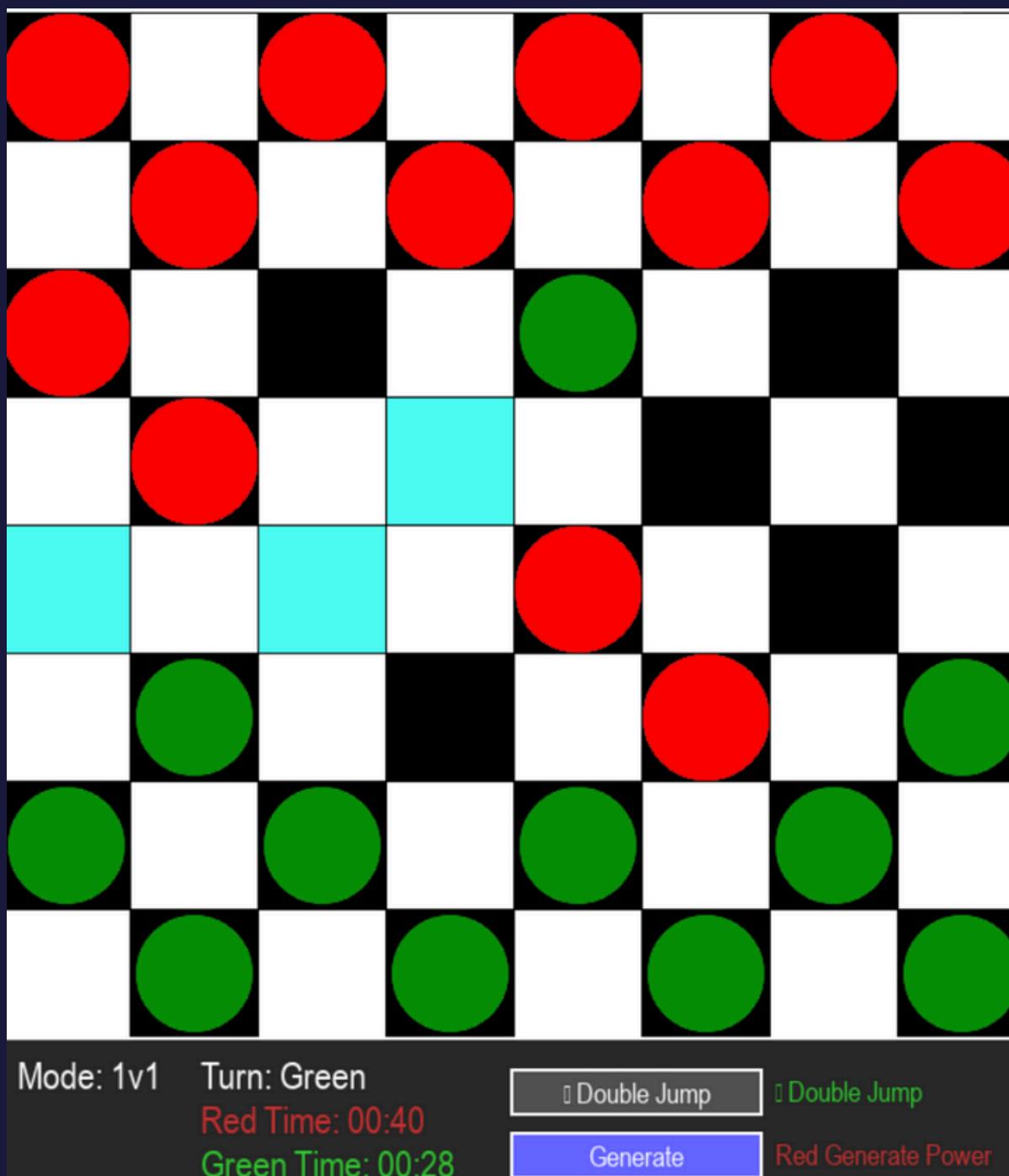
**הבא של היריב**



**השחקן הירוק לא  
יכול לאכול שחקן  
אדום בתור הזה**

# כללי המשחק

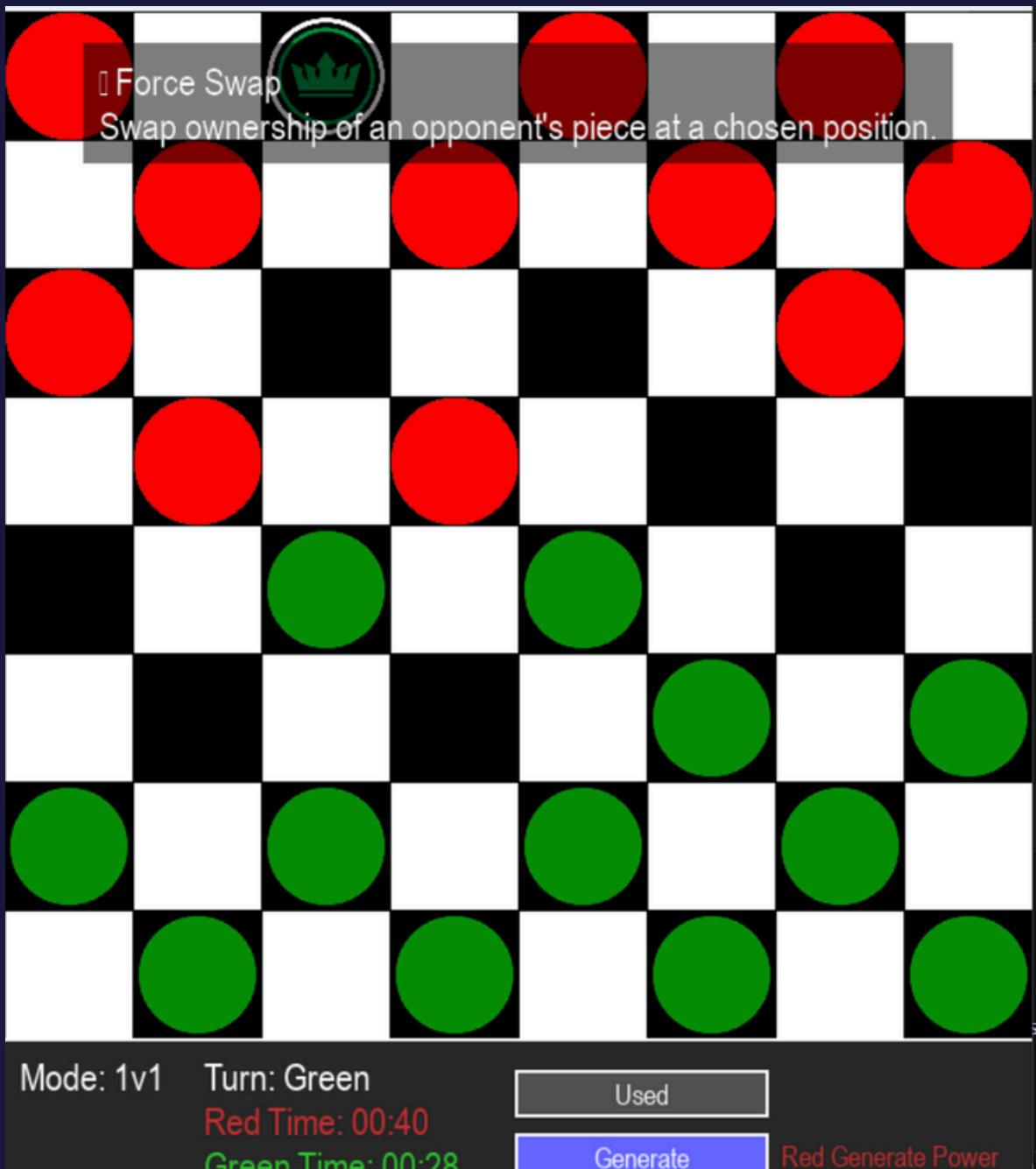
- **שימוש בכוחות מיוחדים: DOUBLE\_JUMP:**  
**שימוש בכוח זה מאפשר למשתמש קפיצה של 2 ריבועים**



# כללי המשחק

- שימוש בכוחות מיוחדים:  **FORCE\_SWAP**

**שימוש בכוח זה מאפשר למשתמש החלפת שחקו מהיריב**



**אם המשתמש בוחר  
שחקו שנמצא  
בבסיס היריב אז  
הוא הופך למילר  
למשתמש**

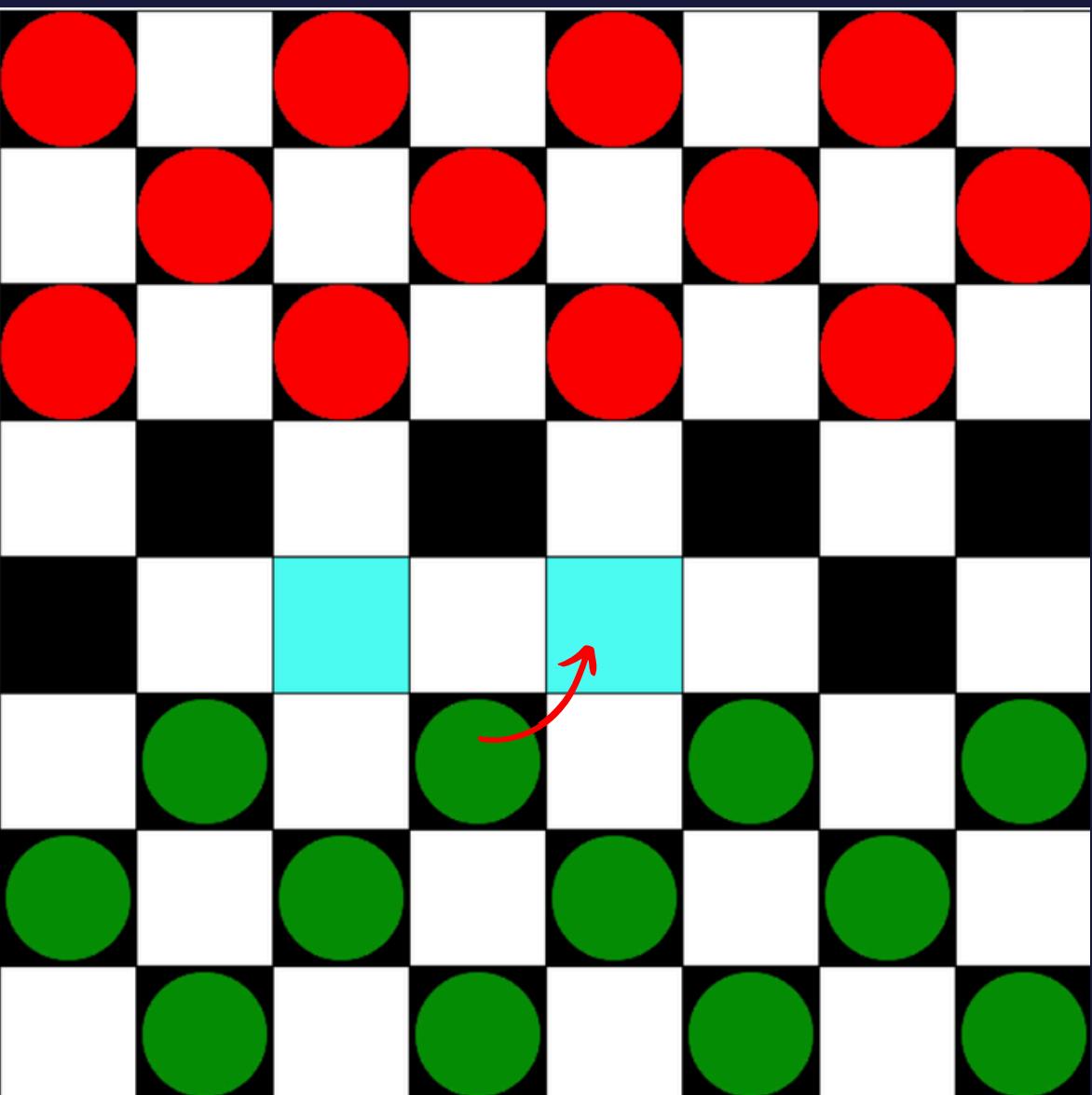
# כללי המשחק



• שימוש בכוחות מיוחדים: MOVE\_TWICE

שימוש בכוח זה מאפשר לשחק שני טורים

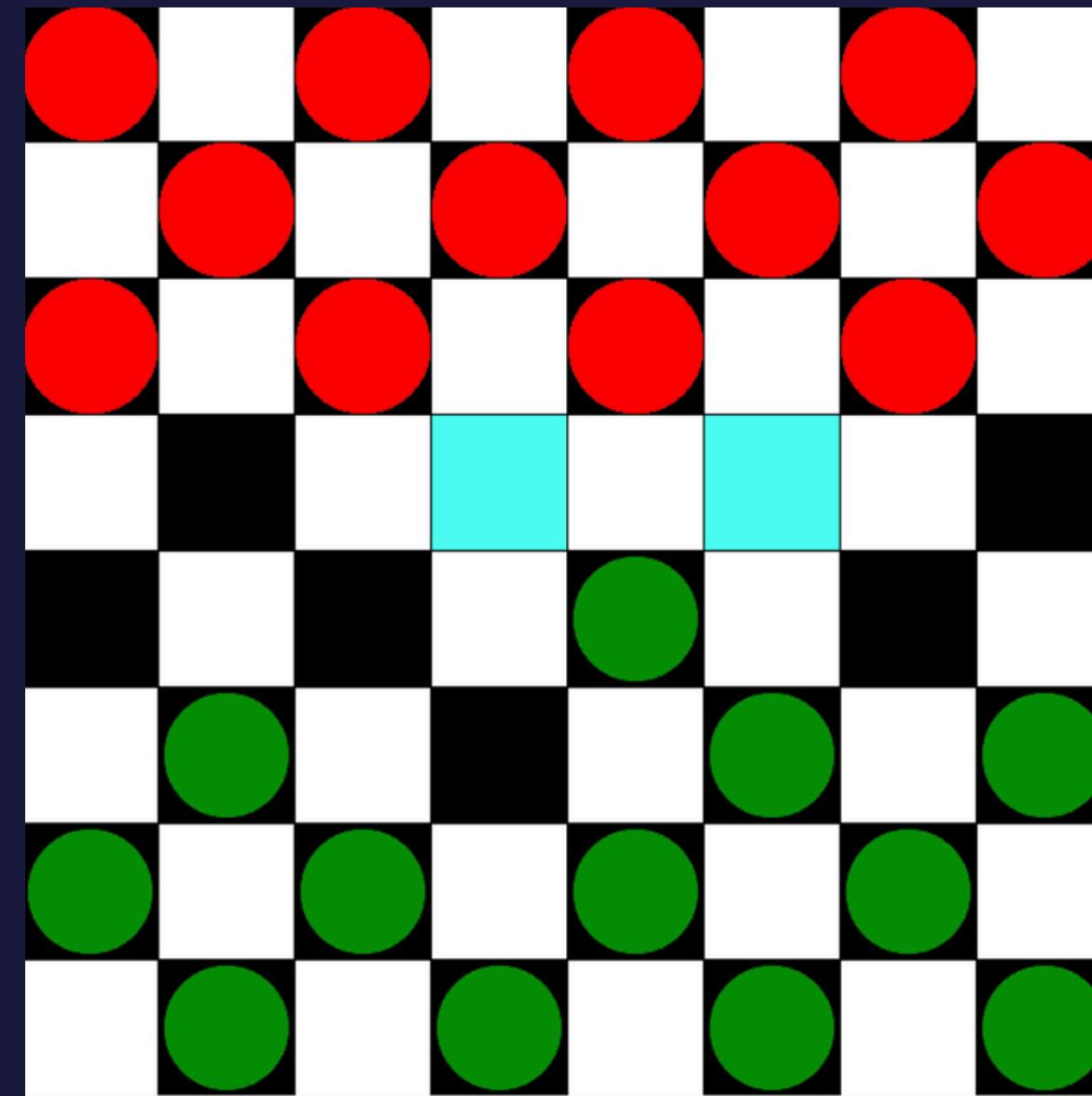
#1



Mode: 1v1 Turn: Green  
Red Time: 00:40  
Green Time: 00:32

Move Twice    Move Twice  
Generate    Red Generate Power

#2



Mode: 1v1 Turn: Green  
Red Time: 00:40  
Green Time: 00:37

Used    Generate  
Red Generate Power

# שימוש באלגוריתם Minimax

- plain\_minimax(board, depth, maximizing, current)
- חיפוש רגיל לכל המסלול עד עומק  $depth$
- כל נוד (node) מייצג מצב לוח; ענפים מייצגים מהלכים
- בסיסים: פונקציית הערכת מחזירה ערך, בעל ערך גבוה  $\leftrightarrow$  טוב ל-*maximizing*

חסרונות:

- סיבוכיות ( $O(b^d)$ ) מלאה – גדל אקספוננציאלית
- איטי מאד עבור עומקים מעל 6-7 בриוות משחק רגיל

# שימוש באלגוריתם Alpha-Beta Pruning

- `minimax_ab(board, depth, α, β, maximizing, current)`
- משלבת חיתוך ענפים בהם הערך אינו יכול לשפר תוצאה קיימת
- $\alpha = \text{הטוב ביותר שמצאנו עד כה עבור Maximizer}$
- $\beta = \text{הטוב ביותר שמצאנו עד כה עבור Minimizer}$
- אם  $\alpha \leq \beta \rightarrow \text{ניתן להפסיק חיפוש בתת-ענף (prune)}$

יתרונות:

- מקטין משמעותית את מספר המוצבים שנבדקים
- מאפשר לחפש לעומקויות גדולים יותר באותו הזמן

# סיבוכיות

$$O(b^d)$$

כasher:

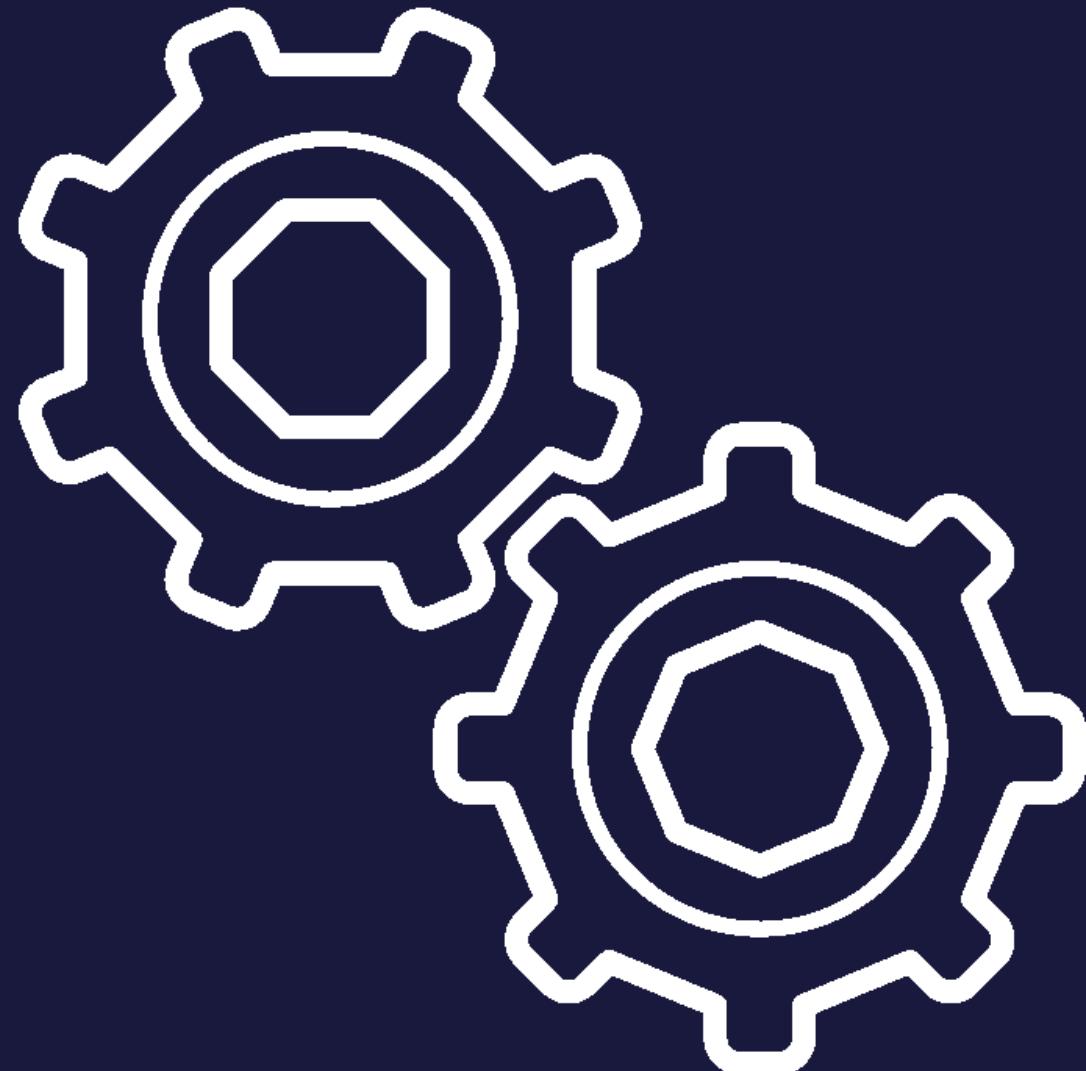
- **b = מוחץ מספר המהלךים האפשריים בכל מצב (branching factor)**
- **d = עומק החיפוש**

במשחק דמקה רגיל (לוח  $8 \times 8$ , בלי כוחות או מלכים מיוחדים),  $10-8 \approx b$ .

במשחק שלנו, בගל:

- לוחות גדולים יותר ( $12 \times 12$  עד  $12 \times 12$ ) → יותר כלים → יותר מהלכים,
- מלci-על שכולים "לעוף" לכל כיוון בכל טווח → מוסיפים עשרה אפשרויות,
- חיילים רגילים שאוכלים אחרה (לא קיימים בדמקה רגילה),
- כוחות מיוחדים שפותחים אפשרויות חדשות (כמו MOVE\_TWICE, FORCE\_SWAP, ...),  
ה-branching factor יכול להגיע ל-20-30 ואף יותר בתור!

# פונקציית הערכתה (Evaluation)

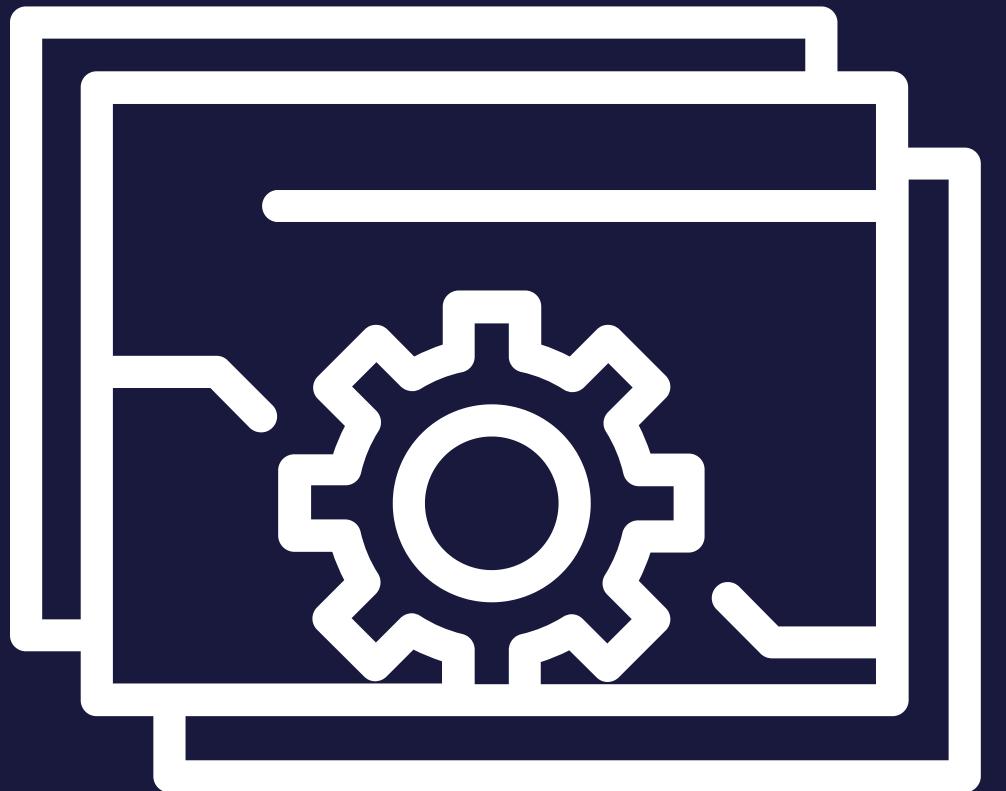


- **כוללת:**
- 1. **חומרה:**
  - $\text{Man} = 1$ ,  $\text{King} = 2$ ,  $\text{Super-King} = 3$
  - חישוב:  $\text{score} = \Sigma(\text{יריב}) - \Sigma(\text{שחקן})$
- 2. **יכולות מיוחדות:**
  - אם **DOUBLE JUMP** פועל  $\leftarrow$  תנוצה אפשרית ארוכה יותר
  - אם **UNITY IMMUNITY** פועל  $\leftarrow$  כלים מוגנים מענישה
- 3. **אכילה לאחור:**
  - חיל רגיל יכול לאכול גם לאחור  $\rightarrow$  מגדיל אפשרויות
- **התאמות:**
- **גודל הלוח** ( $8 \times 8 \rightarrow$  פחות כלים,  $12 \times 12 \rightarrow$  יותר כלים)

**תוצאות:**

- הערכתה כוללת מצב חומרה + עומק אסטרטגי לפי כוחות ו שינויים בחוקי תנוצה

## שפת תכנות



שפה:

Pygame עם ספרייה Python

למה ?Python

- **פשטות וקריאות גבוהה ("קוד כמעט כמו אנגלית")**
- **מהירות פיתוח מהירה - במיוחד לפרויקטים אקדמיים**
- **ספריות עשירות לטיפול ב-(Pygame) GUI ואלגוריתמים**
- **אפשרות להרצה בין מערכות הפעלה בלי שינוי קוד**

# אתגרים - מה היה קשה לימוש

אינטגרציה של כוחות מיוחדים + AI

- **שימוש (DOUBLE\_JUMP, IMMUNITY, MOVE\_TWICE)()** לצד `should_ai_use_power()` כדי שה-AI יקבל החלטה רנדומלית ( $\approx 30\%$ ) מתי להשתמש בכוח התאמת פונקציית ההערכתה: הכוח היקר ביותר לפי מצב הלוח (מספר כלים, עמדת אסטרטגיית)
- **שילוב בזרימת המשחק** כך שכח מופעל רק פעם אחת לכל משחק, ועדין בזמן הנכון

# אתגרים - מה היה קשה למשימה

החלטת AI מתי וair להשתמש בכוח:

- הוספנו בדיקת “pieces < opponent\_pieces” או “#ai\_pieces < 8#” כטיריגר
- פונקציית הערכה תשפייע מתי מחליטים להפעיל כוח מיוחד
- שימוש מלך-על (Super-King)
- תנועת “מעופף” + אכילה בכל מרחק → מרחב חיפוש רב מאד
- שמירה יעילה של מהלכי טוס (בעיקר DFS לעומק רב)
- השפעה על פונקציית הערכה (ערך עלות/תועלת גבוהה יותר למלך-על)

# אתגרים - מה היה **קשה** לימוש



אתגרים נוספים:

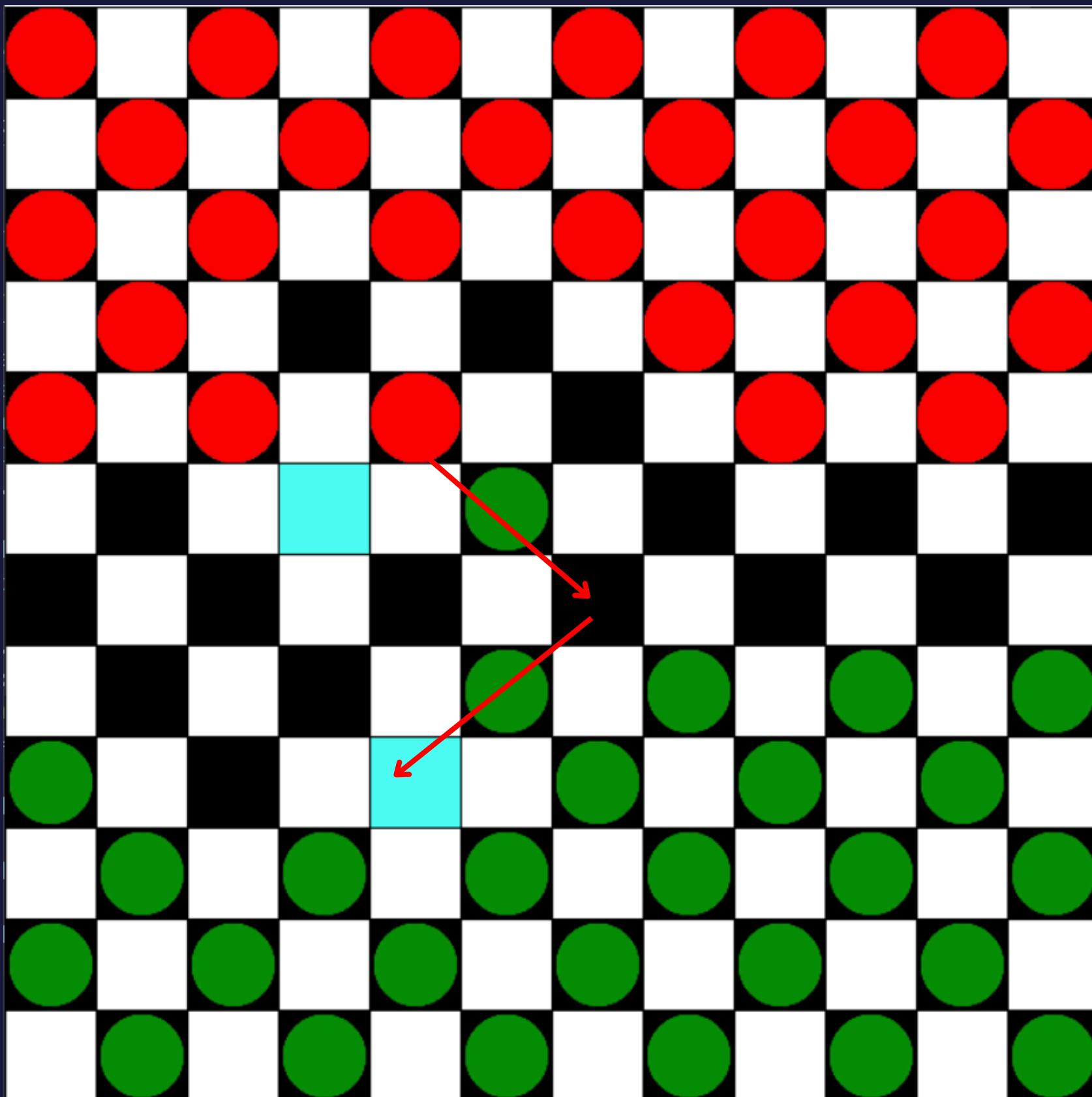
- אופטימיזציה וביצועים בשילוב عمוקי חיפוש גבויים ((<sup>p</sup>b)O)
- ניהול אינזיות מרובות-שלבים (פסיקות אחרי כל אכילה)
- סנכרון טימרים ו-IO (לחצן כוח אחד למשחק, קצב עדכון מסר)
- תמיכה בשלושה גגלי לוח שונים בלי שינוי מהותיים בלוגיקה



**Note:**

**לא היו שינוי  
יחסית להצעת  
הפרויקט שהיגשו**

# כמה תמונות על משחק

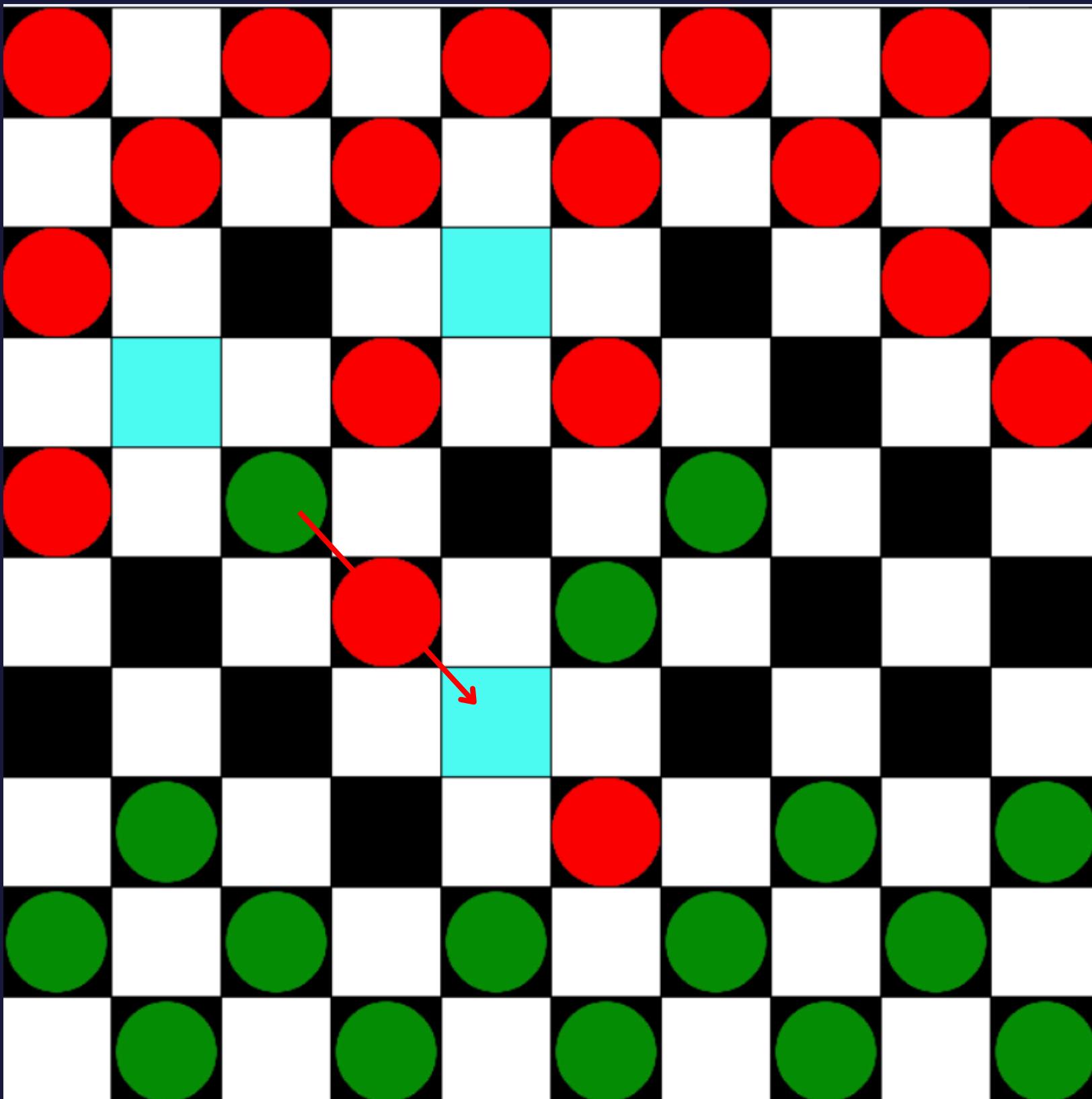


Mode: 1v1 Turn: Red  
Red Time: 00:31  
Green Time: 00:40

Generate Green Generate Power  
Generate Red Generate Power

אכילה של שני שחקנים בלוח 12x12

# כמה תמונות על משחק

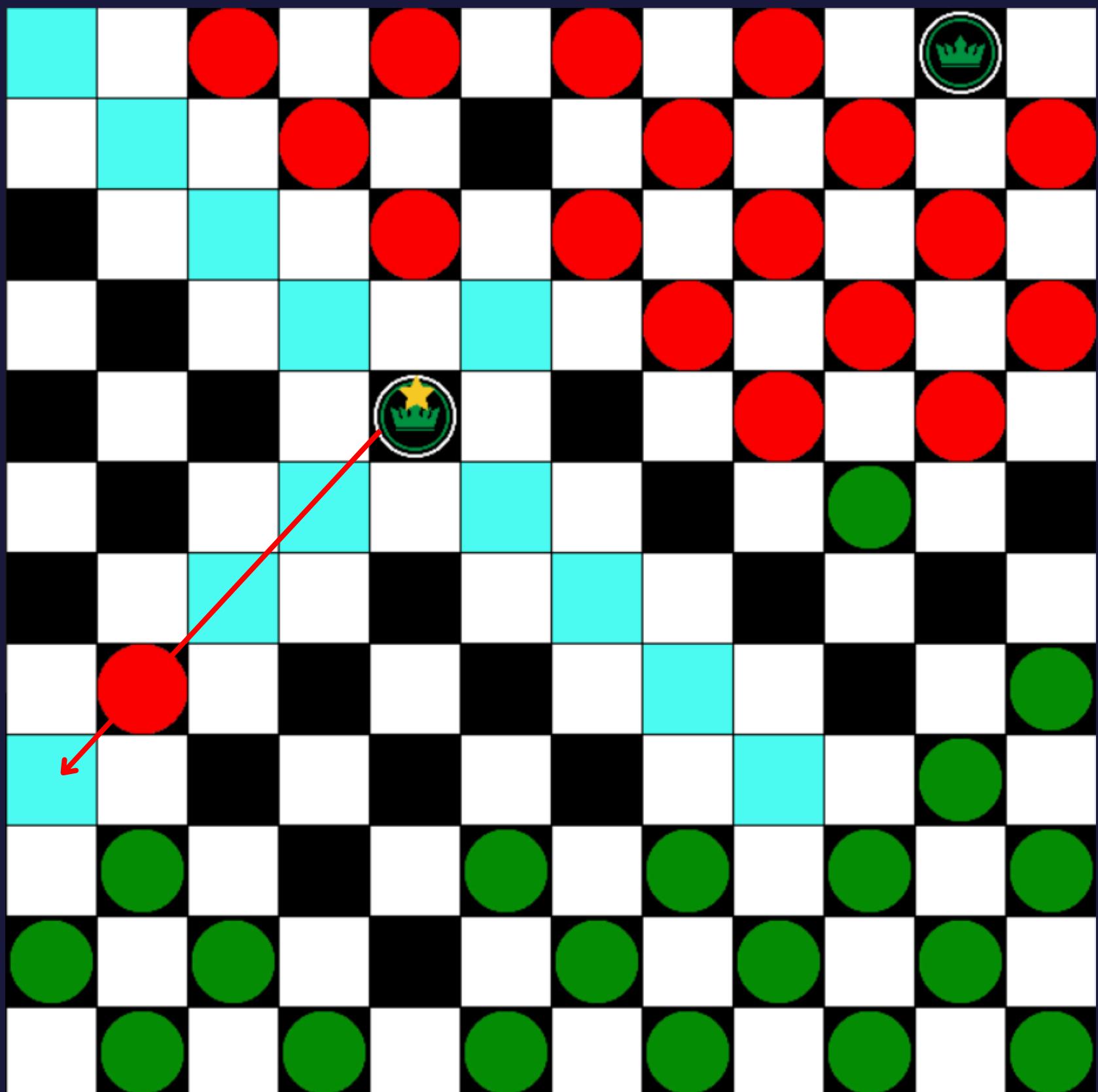


Mode: 1v1 Turn: Green  
Red Time: 00:40  
Green Time: 00:34

Generate Green Generate Power  
Generate Red Generate Power

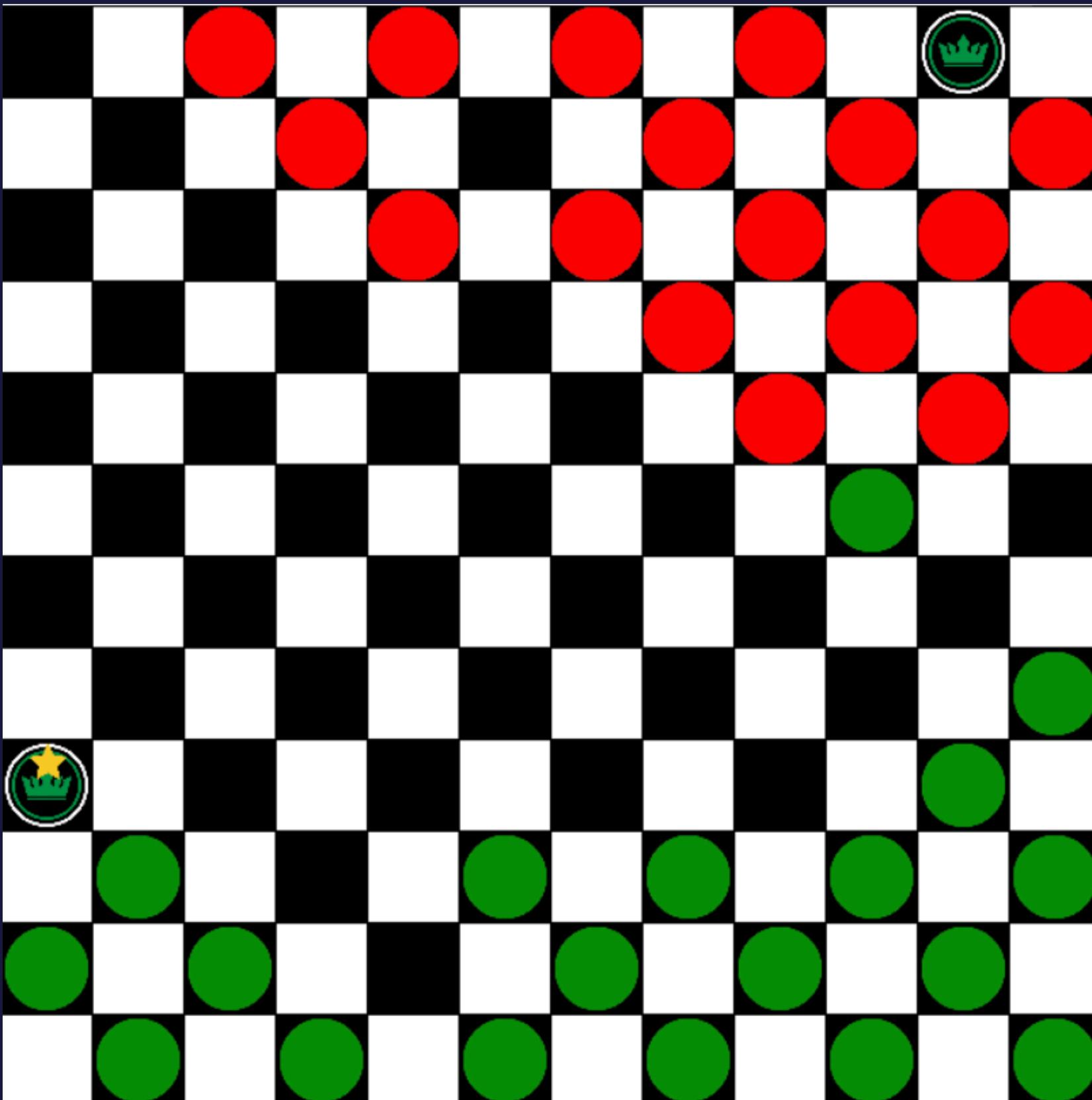
אכילה לאחורה של שחקן אדום בלווי 10x10

# כמה תמונות על משחק



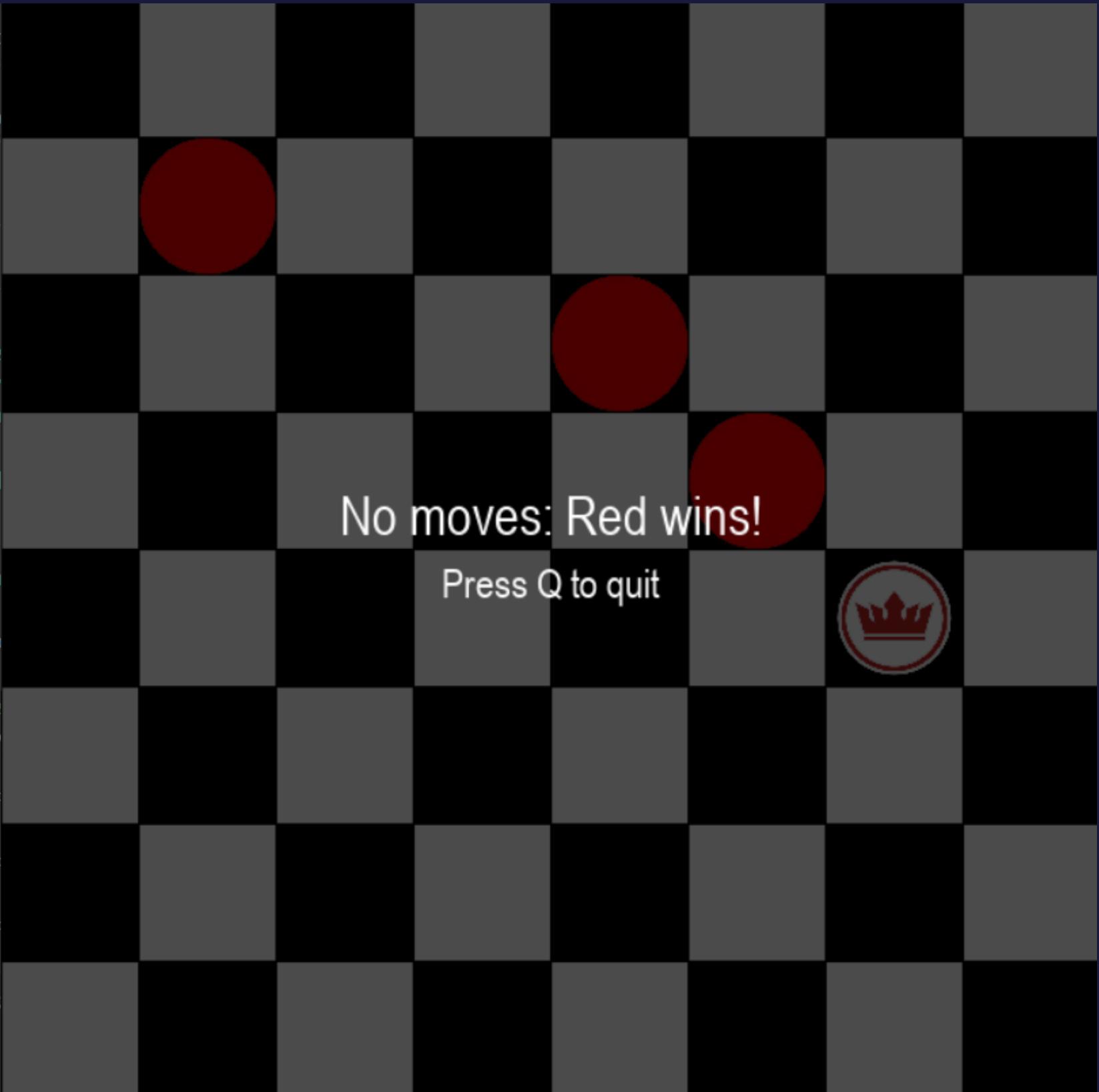
דוגמה למלכית מלך על  
למשל: לפि חץ אדום

# כמה תמונות על משחק

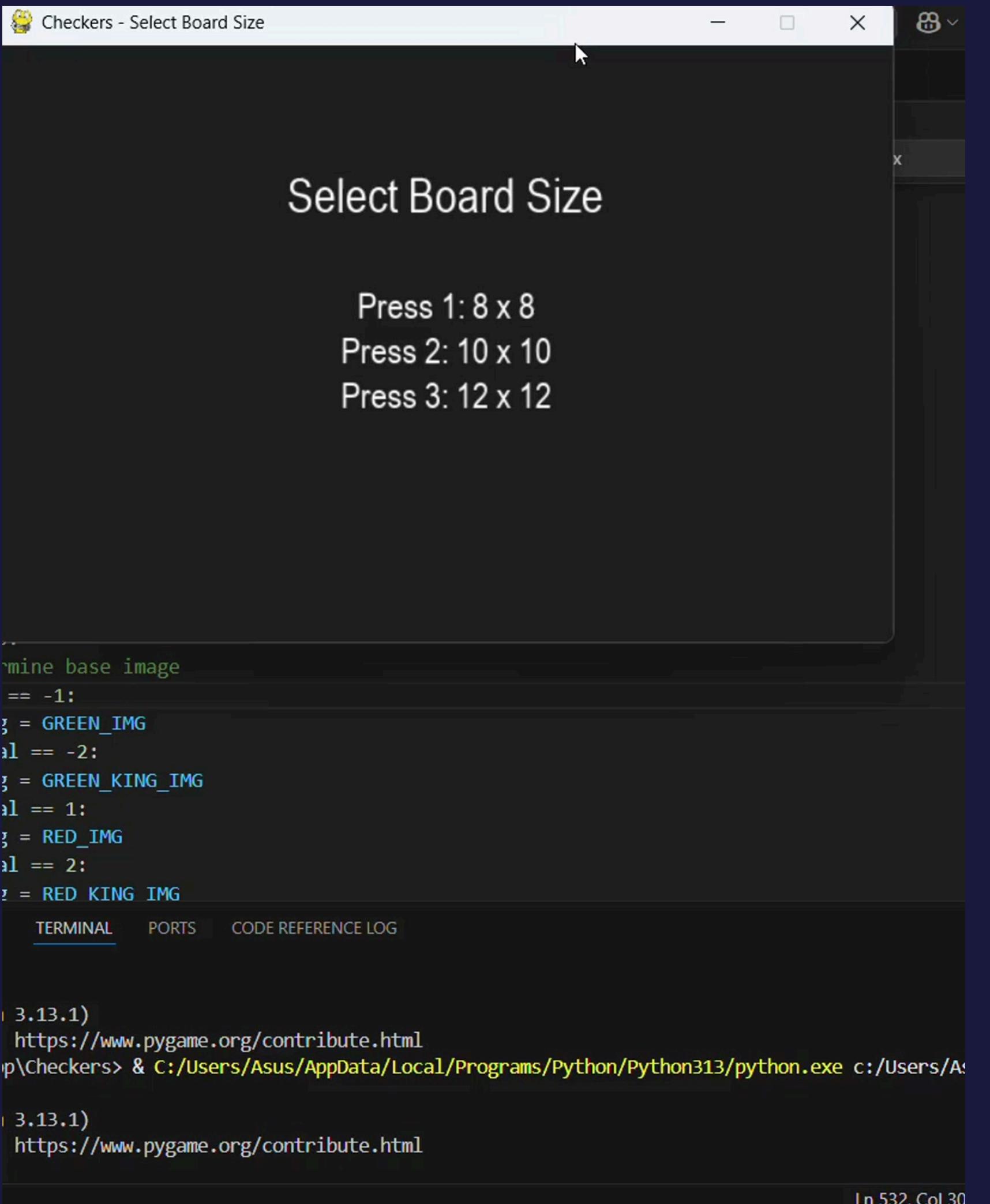


דוגמה למלכי מלך על  
למשל: לפי חץ אדום

# כמה תמונות על משחק



**ניצחון של שחקו בצבא אדום**



# כמה תמונות על משחק

הדגמה של מחשך נגד AI עם אלגוריתם  
Depth Alpha-Beta Pruning



Thank you