ספרון פרוייקט מעבדה:

מגיש :

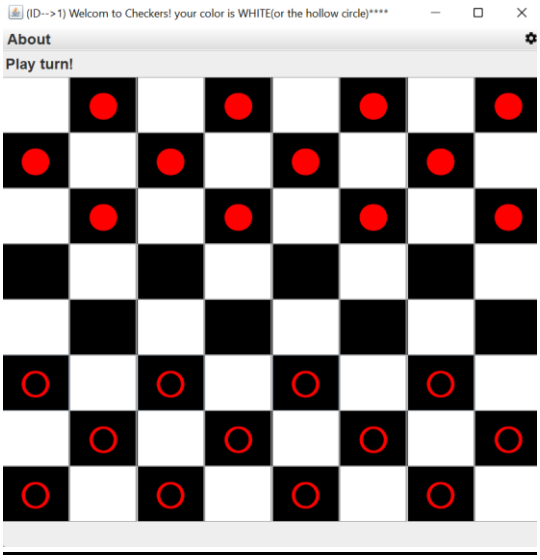
נתן צוברי נ"י.

תז' – 326616703

משחק רשת דמקה המכיל שחקן AI.

תיאור המשחק:

דמקה – משחק לוח בגודל 8X8 שמתנהל בין 2 שחקנים.
שחקן אחד משחק עם הכלים השחורים ואחד עם הלבנים.
הלוח ההתחלתי מסודר כך ש12 כלים ממוקמים על 3 השורות הראשונות של הלוח במשבצות השחורות, ועל שלושת האחרונות ממוקמים 12 הכלים הלבנים.
השחקן בעל הכלים הלבנים מתחיל ראשון לשחק, מטרת המשחק היא לאכול את כל הכלים של היריב, כאשר הראשון שמצליח מנצח. במידה ויש לשני השחקנים אותה כמות מלכים ובמשך 5 תורות לכל אחד לא קרה כלום (כלי נאכל או מישהו ניצח) המשחק יסתיים בתיקו.
לוח משחק התחלתי נראה כך :

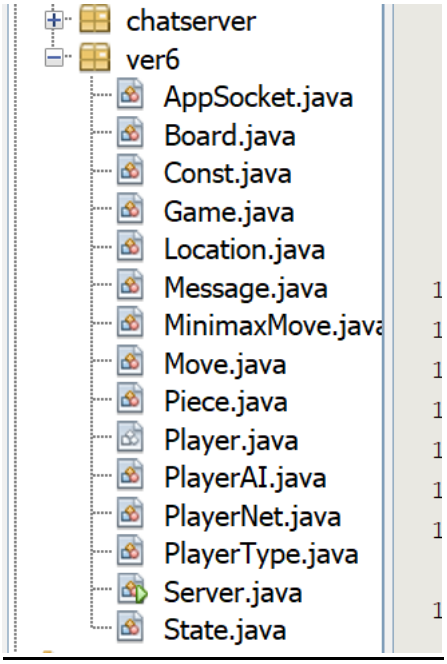


כל שחקן בתורו לוחץ על כלי שברצונו להזיז. לאחר הלחיצה המשבצות שאליהם השחקן יכול לזוז יצבעו באפור. כלי לבן יכול לנוע למשבצות השחורות הסמוכות באלכסון שמעליו וכלי שחור יכול לנוע למשבצות השחורות הסמוכות באלכסון מתחתיו (כלים לבנים נעים לכיוון מעלה הלוח והשחורים לכיוון מטה).
אכילה - כאשר באחת המשבצות באלכסון עומד כלי יריב והמשבצת הבאה באלכסון ריקה, ניתן לעבור אליה ובכך לאכול את הכלי היריב (להוציא אותו מהמשחק).
מלכים - אלו כלים שהצליחו להגיע לקצה השני של הלוח (השחורים למשבצת הכי תחתונה בלוח והלבנים לעליונה ביותר) והם יכולים לנוע לאורך כל האלכסונים שלהם עד סוף הלוח או עד שיש שני כלים יריבים אחד אחרי שני באלכסון.

מרכיבי הפרוייקט:

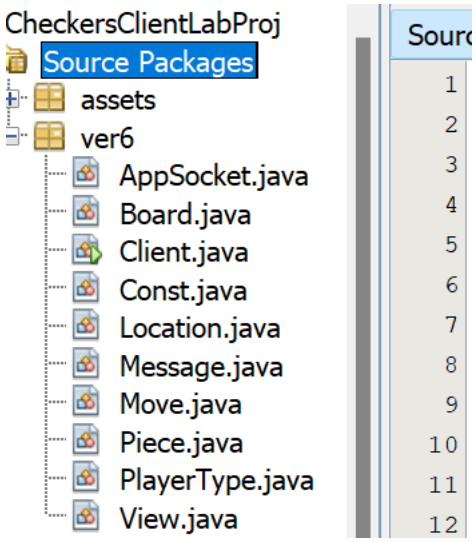
הפרוייקט מורכב מצד שרת וצד לקוח,

צד השרת בנוי מהמחלקות הבאות:



כאשר ה SERVER מכיל את המיין.

וצד הלקוח בנוי מהמחלקות הבאות:



כאשר ה CLIENT מכיל את המיין.

פירוט על המחלקות בשרת :

1. **AppSocket-** המחלקה מייצגת את שקע התקשורת (ה- Socket) המשמש לתקשורת בין השרת ללקוחות. המחלקה מורכבת מ-2 שקעים :

msgSocket שתפקידו לשמש עבור העברת נתונים מהשרת ללקוח וכן מהלקוח לשרת (בעיקר דברים שקשורים למשחק כמו מהלך שנעשה),

ו- cmdSocket מטפל בהתנתקויות.

2. **Board-** מחלקה המייצגת את הלוח המשחק הלוגי. כאשר מבצעים מהלך הלוח הלוגי מתעדכן ונשלח ללקוחות ובעזרתו הלוח שלהם מתעדכן לאחר כל מהלך.
3. **Const-** מחלקת הקבועים. משמשת בהעברת ההודעות לשני הצדדים כנושא ההודעה לפיה הצד שמקבל את ההודעה יודע כיצד להגיב.
4. **Location-** מחלקה המייצגת מיקום כלשהו בלוח.
5. **Move-** מחלקה המייצגת מהלך מסויים. מורכב מ- 2 מיקומים : מיקום המקור של הכלי ומיקום היעד.
6. **PlayerType-** מחלקה המתארת את הנתונים של השחקן שהתחבר (ID של השחקן והאם הוא בחר לשחק נגד שחקן מחשב.
7. **Message-** מחלקה המייצגת הודעות העוברות בין השרת ללקוחות. ההודעה כוללת **תמיד** את נושא ההודעה (const) ועוד נתונים.
8. **Piece-** מחלקה המייצגת כלי על הלוח הלוגי. (את סוג הכלי KING או לא וכו'.
9. **Game-** מחלקה המייצגת משחק בין 2 שחקנים- אנושי מול אנושי או אנושי מול ממוחשב
10. **MinimaxMove-** מחלקה המייצגת מהלך מינימקס עם עומק וניקוד. המחלקה יורשת מהמחלקה Move .

11. **Player-** מחלקה אבסטרקטית (לא ניתן ליצור ממנה עצם- מחלקה כללית עבור תת מחלקות כלשהם). מייצגת שחקן כלשהו במשחק (אנושי או ממוחשב).
12. **PlayerAI-** מחלקה המייצגת שחקן AI ממוחשב. המחלקה יורשת מהמחלקה האבסטרקטית Player .כאשר שחקן אנושי כלשהו בוחר לשחק נגד AI, נוצר עצם מהמחלקה, מולו משחק השחקן האנושי.
13. **PlayerNet-** מחלקה המייצגת שחקן אנושי. המחלקה יורשת מ- Player.
14. **State-** מחלקה המחזיקה בשחקן שתורו לשחק ולוח המשחק הנוכחי, המחלקה אחראית על הלוגיקה של המשחק .
15. **Server-** מחלקה המייצגת את שרת המשחק ודרכה מריצים את השרת אליו יתחברו הלקוחות.

פירוט על המחלקות בצד הלקוח :

1. **AppSocket-** המחלקה מייצגת את שקע התקשורת (ה- Socket) המשמש לתקשורת בין השרת ללקוחות. המחלקה מורכבת מ-2 שקעים :

msgSocket שתפקידו לשמש עבור העברת נתונים מהשרת ללקוח וכן מהלקוח לשרת (בעיקר דברים שקשורים למשחק כמו מהלך שנעשה),

ו- cmdSocket מטפל בהתנתקויות.

2. **Board-** מחלקה המייצגת את הלוח המשחק הלוגי. כאשר מבצעים מהלך הלוח הלוגי מתעדכן ונשלח ללקוחות ובעזרתו הלוח שלהם מתעדכן לאחר כל מהלך.
3. **Const-** מחלקת הקבועים. משמשת בהעברת ההודעות לשני הצדדים כנושא ההודעה לפיה הצד שמקבל את ההודעה יודע כיצד להגיב.
4. **Location-** מחלקה המייצגת מיקום כלשהו בלוח.
5. **Move-** מחלקה המייצגת מהלך מסויים. מורכב מ- 2 מיקומים : מיקום המקור של הכלי ומיקום היעד.
6. **PlayerType-** מחלקה המתארת את הנתונים של השחקן שהתחבר (ID של השחקן והאם הוא בחר לשחק נגד שחקן מחשב.
7. **Message-** מחלקה המייצגת הודעות העוברות בין השרת ללקוחות. ההודעה כוללת **תמיד** את נושא ההודעה (const) ועוד נתונים.
8. **Piece-** מחלקה המייצגת כלי על הלוח הלוגי. (את סוג הכלי KING או לא וכו'.
9. **Client-** מחלקה המייצגת לקוח. מריצים את המחלקה ודרכה מתחברים לשרת ומתחילים לשחק. צד הלקוח מעביר הודעות לשרת ומקבל הודעות ממנו ומעדכן את לוח המשחק בהתאם

10. **View-** מחלקה המשמשת לתצוגה (GUI) של המשחק. המחלקה אחראית על תצוגת המשחק ומתעדכנת בהתאם להודעות שמתקבלות מהשרת.

מדריך למשתמש:

הסבר מפורט הממחיש משחק מהתחלתו ועד סופו על ידי תמונות והסברים.

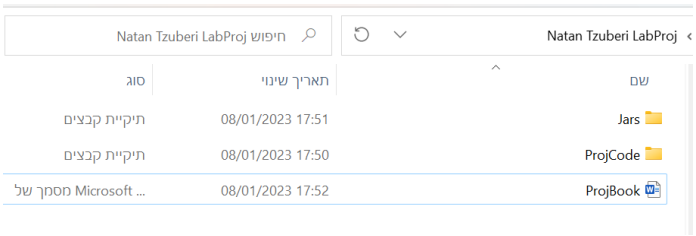
דרישות טכניות

קוד התוכנה הורץ ונבדק על מחשבים עם המאפיינים הבאים, שהם גם הדרישות הטכניות להרצת קבצי הפרויקט שעליהם :

1. התקנת Java Runtime Environment בגרסה 8 ומעלה.
2. קישוריות לרשת ופורטים פתוחים.

הרצת המשחק והנחיות שימוש

בתיקיית הפרוייקט המצורף יש 3 תת-תקיות :



יש להיכנס לתת-התיקייה **jars** שבתוכה ישנם 2 קובצי הרצה :

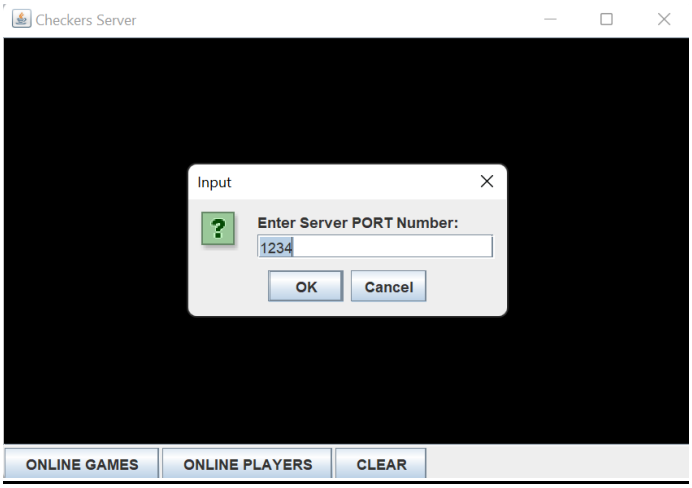
שם	תאריך שינוי	סוג
CheckersClientLabProj	08/01/2023 05:10	Executable Jar File
CheckerServerLabProj	08/01/2023 05:10	Executable Jar File

בכדי להריץ את המשחק נפעיל קודם את **השרת** ע״י הרצת הקובץ **Server.jar**, שמיד ממתין לחיבור לקוחות מריצים את הSERVER פעם אחת בלבד!

כעת מפעילים את **הלקוח** ע״י הרצת הקובץ **Client.jar** שמייצג שחקן-רשת יחיד. ניתן להריץ **מספר לקוחות** כדי ליצור מספר שחקנים. עבור כל שני שחקנים, השרת מייצר משחק.

הדגמת הרצת משחק על ידי צילומי מסך:

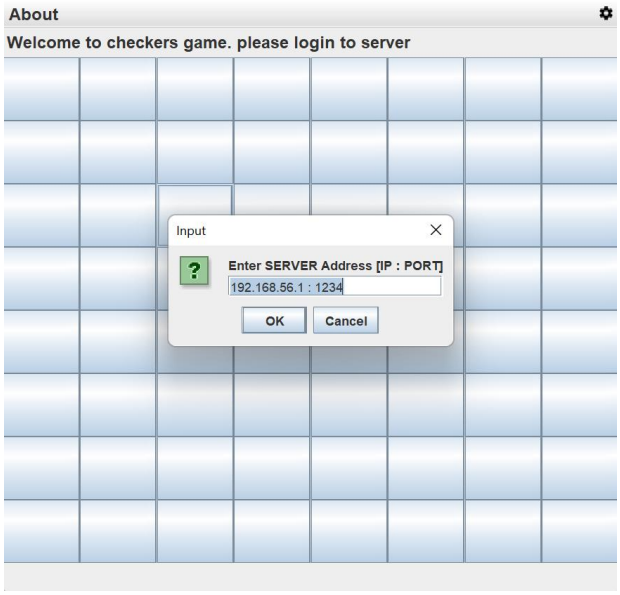
על מנת להפעיל את השרת נלחץ על קובץ ה-jar. עם הלחיצה יתקבל החלון הבא:



מתקבל חלון input שבו נמלא את הפורט הרצוי.

הרצת הלקוח :

נריץ את קובץ ההרצה של ה- client לאחר ההרצה יתקבל המסך הבא:



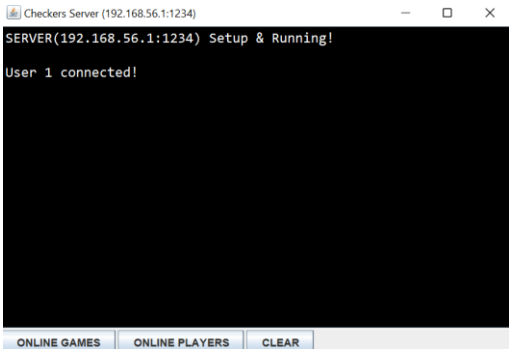
בחלון ה- input שקופץ יש להכניס את ה- IP והפורט עליהם נמצא השרת.

לאחר שהכנסנו את הפרטים יקפוץ החלון הבא :

בחלון הזה נבחר את סוג השחקן נגדו אנו רוצים לשחק.

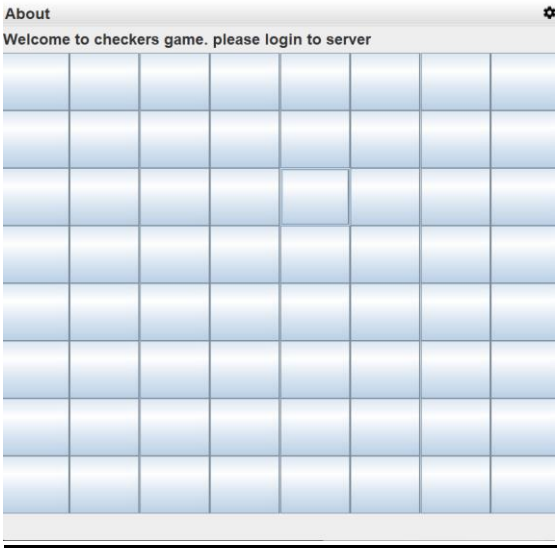
<u>יציאה-</u> במידה והמשתמש החליט שלא להתחבר לשרת אלא להתנתק החלון יסגר.

לאחר שהשחקן מתחבר תתקבל הודעה בשרת :



במידה ובחרנו לשחק נגד שחקן אנושי אחר (לא סומנה תיבת play with AI)

ואין שחקן אנושי אחר שמחכה, מסך המשחק יפתח על false עד ששחקן אחר יתחבר:



לאחר שיתחבר עוד שחקן אנושי נוסף או שהתחברנו ומישהו כבר המתין במקרה שהמשתמש בחר במשחק נגד שחקן מחשב המשחק יתחיל מיד.

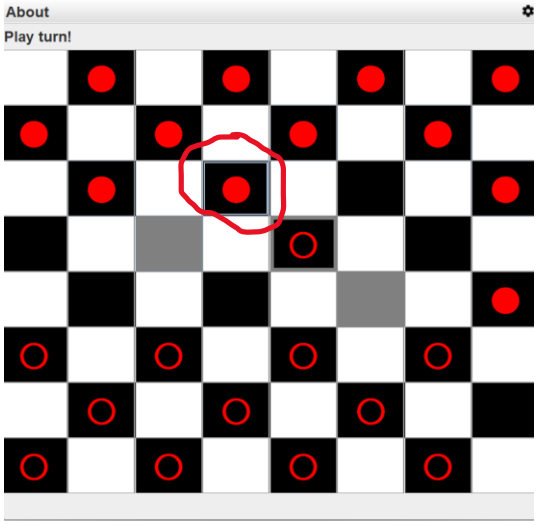
3.3.3 הפעלת המשחק וממשק משתמש גרפי (GUI)

לאחר שמתחיל המשחק השחקן הלבן יהיה ב - play turn והשחקן השחור יהיה ב – wait turn

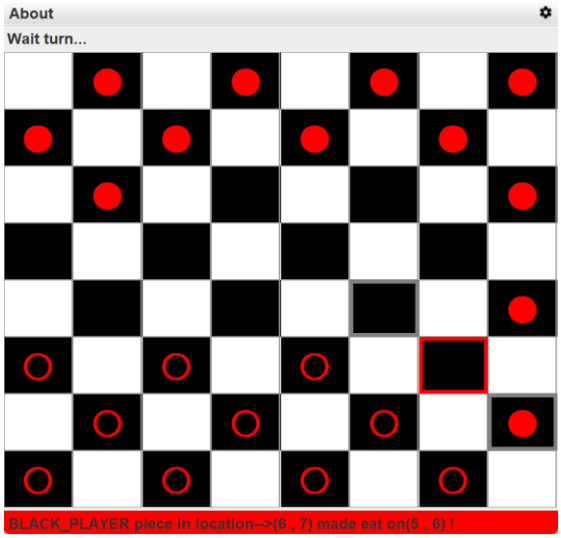
השחקן ילחץ על החייל שאותו הוא רוצה להזיז והמהלכים החוקיים יסומנו באפור השחקן ילחץ על המשבצת שברצונו לזוז אליה.

במידה והשחקן יחליט לשחק עם חייל אחר הוא יצטרך ללחוץ על החייל שוב (שהסימונים האפורים יעלמו)

דוגמא :



חייל בתמונה יש שתי אופציות לאכול או לא (במידה ולא יאכל החייל ישרף!)
במידה והשחקן בחר לאכול תוצג הודעה בתחתית הלוח שתודיע על החייל שאכל ועל החייל שנאכל
המשבצת ש"נאכלה" תצבע באדום :



סיום משחק:

ניצחון - יתקבל כאשר לשחקן אחד נותרו חיילים ולשני נגמרו כולם.

תיקו - יתקבל כאשר לשני השחקנים יש אותה כמות מלכים ובמשך 5 תורות לא התפתח כלום במשחק (לא נאכל חייל או מישהו ניצח).

לאחר שנגמר המשחק תתבצע ספירה לאחור של 7 שניות ולאחר שתסתיים יתחיל משחק חדש בין 2 השחקנים.

במידה ובאמצע משחק אחד השחקנים התנתק מכל סיבה שהיא (נפילה או מכל סיבה אחרת) השחקן היריב מקבל הודעה שנגמר המשחק עקב התנתקות היריב והוא זוכה בניצחון טכני:

תפריטים: קיימים 2 תפריטים עבור כל שחקן:

About- תפריט הסבר על המשחק ובו 3 אפשרויות:

1. **game & rules -** הסבר על המשחק והחוקים שלו
2. **credits -** קרדיטים לאלו שעזרו בבניית הפרויקט
3. **Quit** – במידה ושחקן רוצה לפרוש.

Settings - שינויים גרפיים בלוח המשחק:

1. Sound effect -מוסיף sounds למשחק (לא הספקתי לטפל בזה ולכן כרגע האופציה לא עובדת.
2. Spinner - Set win size המאפשר לשחקן לשנות את גודל לוח המשחק.