**עבודות גמר**

**נושא הפרויקט: "Queah"**

**מגיש: נתנאל חכמון**

**ת.ז:** 324199504

**מנחה: מריו סולאי**

**תאריך הגשה: 1.5.2022**

**תוכן**

[תקציר: 3](#_Toc101699917)

[תיאור הנושא: 3](#_Toc101699918)

[רקע תאורטי: 4](#_Toc101699919)

[תיאור הבעיה האלגוריתמית: 5](#_Toc101699920)

[סקירת אלגוריתמים בתחום הבעיה: 5](#_Toc101699921)

[אסטרטגיה 6](#_Toc101699922)

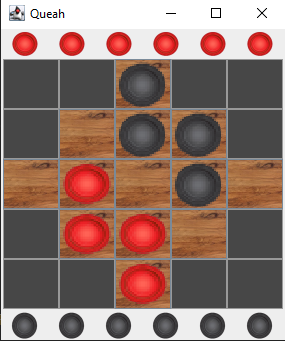
[מבנה נתונים 8](#_Toc101699923)

[תרשים מחלקותUML: 10](#_Toc101699924)

# תקציר:

הפרויקט שבחרתי לעשות הוא queah. זהו משחק בו ניתן לשחק מול המחשב או מול שחקן אחר. בפרויקט זה שילבתי את הנושא תכנות מונחה עצמים ומימשתי אלגוריתמים שונים. בנוסף, המשחק בנוי על אסטרטגיה שיצרתי ומאפשר לשחק בלוחות בגדלים שונים.

# תיאור הנושא:

**לוח המשחק**: המשחק משוחק על לוח מרובע משופע או אלכסוני עם 13 רווחים בלבד.

**כללי משחק הבסיסיים:**

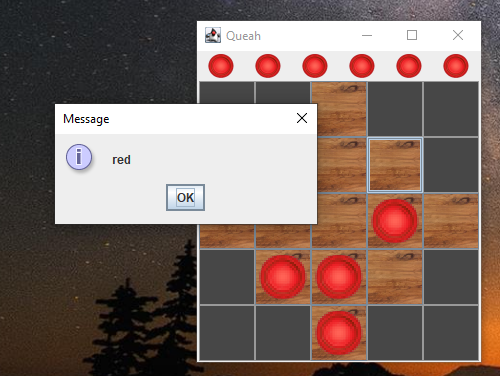
משחק ל – 2 משתתפים, 10 אבני משחק לכל משתתף 4 אבנים על הלוח.

מטרה: "לאכול" את כול האבנים של המשתתף השני.

תנועה על-גבי הלוח: כול שחקן בתור שלו יכול להזיז אבן למקום הצמוד הפנוי אל אותו אבן. שחקן יכול לאכול חתיכת אויב בקפיצה קצרה בגודל של שתיים. הכלי של השחקן חייב להיות צמוד לכלי האויב, ולנחות על שטח פנוי בצד השני. הלכידה חייבת להיעשות בכיוון אורתוגונלי בהתאם לעיצוב המלוכסן או האלכסוני של הלוח. ניתן ללכוד רק חתיכת אויב אחת בכל תור. חתיכה שנתפסה מוסרת מהלוח.

מהלך המשחק:

שחקנים מחליטים באילו צבעים לשחק, ומי מתחיל ראשון. אם כלי של שחקן נתפס, אז השחקן בתחילת התור הבא שלו חייב לקחת חתיכה אחת מהרזרבה שלו, ולשחרר אותו על כל מקום פנוי. יש להחזיר תמיד את מספר הכלים של שחקן על הלוח לארבעה, אלא אם כן השחקן מיצה את הרזרבה שלו. ששחקן יכול להוריד רק חתיכה מהרזרבה שלו, אם אחד הכלים שלהם נכבש בתורו האחרון של היריב.

**סיום המשחק:**

שחקן אדום ניצח

בגלל שלשחור לא

נישאר חיילים-

 שחקן שחור ניצח

בגלל שלאדום לא

נישאר חיילים-

# רקע תאורטי:

Liberian Queah הוא משחק אסטרטגיה מופשט לשני שחקנים מליבריה. השם הרשמי של המשחק הזה אינו ידוע, מכיוון שהוא לא צוין כאשר המשחק נכתב לראשונה. המשחק הוקלט לראשונה בשנת 1882, שחקניו היו חברים בשבט הקואה בליבריה. הלוח המסורתי עשוי מסורג של זרדים, ומקלות יוצרים את החלקים. את החלק העליון של המקלות חותכים בצורה מלוכסנת מצד אחד, ונקראים "גברים", ואילו המקלות של הצד השני נחתכים ישר ונקראים "נשים".

# תיאור הבעיה האלגוריתמית:

במהלך העבודה על הפרויקט נתקלתי במספר בעיות אלגוריתמיות אותן הייתי צריך לפתור כדי שהמערכת תעבוד כמו שצריך:

בדיקת תקינות של המהלך - כאשר המשתמש מבצעה מהלך המחשב בודק אם הוא תקין.

בניית אלגוריתם לשחקן ממוחשב – ליצור שחקן אשר פועל לפי המהלכים של היריב. יש למחשב אלגוריתם אשר סורק את הלוח שומר את כול המהלכים האפשריים לכול חייל וגם נותן משקל לכול אחד מהחלקים שלו.

מציאת כל המהלכים האפשריים לחייל מסוים- מציאת כל האפשרויות למהלכים עבור אחד החיילים ממצב לוח מסוים.

# סקירת אלגוריתמים בתחום הבעיה:

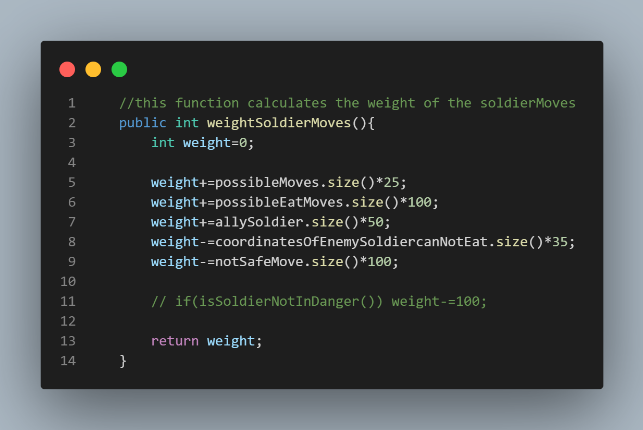
בדיקת תקינות של המהלך - הבדיקות תקינות שהיו הם: אם השחקן מזיז את החייל שלו או שם חייל חדש. אם הוא מזיז אז אם הוא ביצעה אכילה או הזזה רגילה. אם זה אכילה(כאשר יש עד 4 אפשרויות של אכילה. אכילה מתבצעת בקפיצה קצרה בגודל של שתיים ובאמצע יש את היריב) אז האם האכילה הייתה חוקית ואם זה הזזה(כאשר יש עד 4 אפשרויות של הזזה. הזזה מתבצעת בכך שצריך להזיז אבן למקום הצמוד הפנוי אל אותו אבן) אז האם הזזה הייתה חוקית. השחקן שם חייל חדש(בכול מקום רק בלוח) רק אם חייל שלך נאכל בתור הקודם וגם אם נישאר לך חיילים שאתה יכול לשים.

בניית אלגוריתם לשחקן ממוחשב – למחשב יש 3 אפשריות רמה שונות בכול האפשרויות האופציה הראשונה תהייה תמיד לאכול אחר כך אם זה רמה קלה אז הוא יבדוק אם יש חיילים שעומדים להיאכל ואם כן אז הוא יזיז את החייל לאחד מהאפשרויות הזזה הרנדומליות שיש לחייל אם לא אז הוא יבחר חייל בצורה רנדומלית ומזיז אותו לאחד מהאפשרויות הרנדומליות. ברמה בינונית הוא יבדוק אם יש חיילים שעומדים להיאכל ואם כן אז הוא יזיז את החייל לאפשרות הזזה אם המשקל הגבוה ביותר אם אין חייל שעומד להיאכל אז הוא מחשב לכול אחד מהחיילים את המשקל שלו ולוקח את החייל אם המשקל הגבוה ביותר ומזיז אותו לאפשרות אם המשקל הגבוה ביותר. המשקל מתחשב במה שקורה מסביב לחייל ב 2 בלוקים לכול כיוון ( למעלה למטה ימינה ושמאלה ). כאשר זה הרמה הקשה החייל משתמש באלגוריתם Negamax ובוחר את ההזזה לפי אלגוריתם זה.

מציאת כל המהלכים האפשריים לחייל מסוים - מציאת כל האפשרויות למהלכים עבור אחד החיילים ממצב לוח מסוים בכך שמקבליים את קואורדינטות של חייל וזה והאלגוריתם מסורק 2 בלוקים לכול כיוון ( למעלה למטה ימינה ושמאלה ) ובודק אם יש אכילה , הזזה או אין כלום. הוא שומר את כול ההזזות האפשריים ברשימה וכך גם האכילות האפשריים. וזה בודק גם אם החייל יכול להיאכל על ידי היריב.

# אסטרטגיה

האסטרטגיה מתחלקת לכמה חלקים שהם:

1. דירוג החייל של השחקן – דירוג זה מתבצע על מתן נקודות לפי פרמטרים וחישובם כך. כמות אפשרויות הזזה של החייל כפול 25 + כמות חיילים של היריב אשר צמודים לחייל ואפשר לאכול אותם כפול 100 + כמות החיילי בירית הצמודים לחייל כפול 50 – כמות חיילים של היריב אשר צמודים לחייל ואי אפשר לאכול אותם כפול 35 – כמות המהלכים של החייל שיגרמו לו להיאכל על ידי היריב.
2. דירוג הלוח של שחקן – הדירוג מתבצע על ידי חישוב וחיבור הדירוג של כול אחד מהחיילים של השחקן.



1. דירוג הלוח עצמו - הדירוג מתבצע על ידי מתן נקודות לשחקן וליריב שלו על ידי פרמטרים זהים ובסוף להחזיר את ההפרש. פרמטרים הם הדירוג של הלוח השחקן ,כמות הכלים של השחקן , הדירוג של הלוח של היריב ו כמות הכלים של היריב.



1. הוספת חייל חדש – כול אחד מהמקומות בלוח מדורג אם ניקוד קבועה וכאשר מוסיפים חייל חדש עדיף לשים את החייל במיקום הפנוי אם הניקוד הגבוה ביותר.

# מבנה נתונים

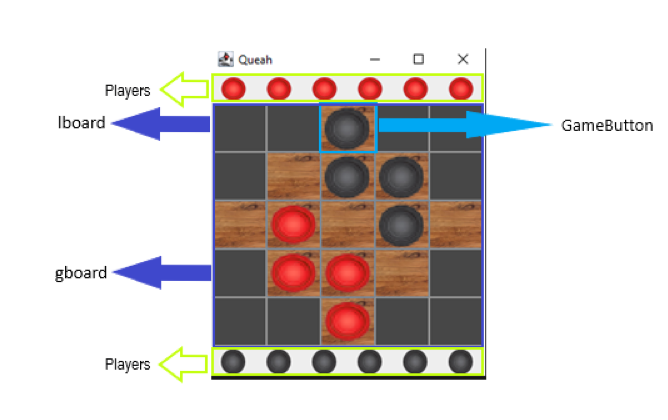
בפרויקט יש כמה מבני נתונים שונים ולכול אחד יש מטרה שונה:

1. יש מטריצה דו ממדית דינמית(שלוש אופציות: קטן ,בינוני וגדול) של int בשם lboard והיא שומרת בכול אחד מהמשתנים במערך את האופציות האלה(1- אם זה לא פעיל/לא קשור ללוח , 0 משבצת ריקה , 1 משבצת אם חייל אדום ו 2 משבצת אם ייל כחול)
2. יש מטריצה דו ממדית דינמית(שלוש אופציות: קטן ,בינוני וגדול) של GameButton(מחזיק תמונה וכפתור) בשם gboard שהיא אחראית לשמירה והצגה גרפית של הלוח.

1. מערך חד ממדי בגודל 8 של int בשם test שימושו היא בשליחת מידע ועדכון מידה כאשר המחשב מבצעה מהלך. כול תא אומר כך.

[newRow][ newColumn][ previsRow][ previsColumn][ eatRow][ eatColum][ isEat][ isSoldierLeft]

1. מחלקה בשם Coordinate אשר שומרת שלוש משנים של int בשם row, column, value. והמטרה שלה היא לשמור את השורה, עמודה ומידע.
2. יש מחלקה בשם SoldierMoves אשר ובתוכו יש מבנה נתונים הללו
   * 1. רשימה של Coordinate בשם possibleMoves ששומרת את כול ההזזות החוקיות של החייל.
     2. רשימה של Coordinate בשם possibleEatMoves ששומרת את כול האכילות החוקיות של החייל.
     3. רשימה של Coordinate בשם notSafeMove ששומרת את כול ההזזות החוקיות אך יגרמו לחייל להיאכל בתור הבה.
     4. רשימה של Coordinate בשם allySoldier ששומרת את כול חיילי ברית אשר צמודים לחייל.
     5. רשימה של Coordinate בשם coordinatesOfEnemySoldiercanNotEat ששומרת את כול החיילים של היריב הצמודים לחייל ואי אפשר לאכול אותם.



כך מתבטאים מבני הנתונים השייכים לגרפיקה.

# תמונה שמכילה טקסט, צילום מסך, מקורה התיאור נוצר באופן אוטומטיתרשים מחלקותUML: