**עבודת גמר**

**לקבלת תואר טכנאי**

**הנדסת תוכנה**

**נושא הפרויקט:**

**שפת תכנות:** Java

**סביבת העבודה:** ‎Intellij

**שם המגיש:** גיל בן בסט

**ת"ז:** 206395873

**מכללה:** אורטחולון

**כיתה:** י"ג 2, מגמת הנדסת תוכנה

**מנחה:** חן אולמר ואופק ראובני

**מאי 2018 תשע"ח**

**תוכן עניינים**

**הצעת פרויקט.........................................................................................................** 5-3

1. **מבוא .............................................................................................................** 6
   1. **מטרה .............................................................................................** 6
   2. **תיאור המערכת.................................................................................** 6
   3. **מבנה המסמך ..................................................................................** 6
   4. **ביבליוגרפיה.....................................................................................** 6
2. **מפרטי תוכנה .............................................................................................** 12-7
   1. **תיאור כללי ....................................................................................** 7-8
   2. **ממשק אדם מכונה MMI ................................................................** 9-10
   3. **יכולות פונקציונאליות.......................................................................** 11
   4. **תיאור אלגוריתמים .........................................................................** 12

1. **תכנון.........................................................................................................** 13
2. **מדריך למשתמש.........................................................................................**14
   1. **הוראות התקנה ( Setup ) ...........................................................** 14

**הצעת נושא לפרויקטים לטכנאים במגמת הנדסת תוכנה**

להדביק כאן את כל הצעת הפרויקט בשלמותה!!!

**1. מבוא:**

**1.1 מטרה:**

מטרת המשחק להגיע למצב של שח-מט כנגד מלכו של היריב. שח-מט קורה רק כאשר אין ליריב מלהלכים חוקיים **וגם** היריב נמצא במצב של שח. שח קורה כאשר המלך מאויים (ניתן לאכילה במהלך אחד). מהלך נחשב כלא חוקי אם בסופו המלך של מבצע המהלך מאויים.

**1.2 תיאור המערכת:**

המערכת מורכבת מלוח משחק שגודלו 8X8.  
המשחק מאותחל עם שורת כלים מיוחדים בשני צידי הלוח הנקרא שורת הבית ומיד לאחריו שורה נוספת של חיילים.  
לכל כלי יכולות תזוזה המאפיינות אותו- רץ זז באלכסון, צריח בקו ישר, מלכה גם באלכסון וגם בישר, פרש זז שתי משבצות לכיוון ישר מסויים ואז עוד משבצת אחת לאחד הכיוונים המאונכים לכיוון התזוזה הראשוני, חייל זז משבצת אחת קדימה ואוכל באלכסונים שבכיוון תנועתו, מלך זז משבצת אחת לכל כיוון שמסביבו.

**1.3 מבנה המסמך:**

הספר כולל ארבעה פרקים:

הפרק הראשון- מבוא למשחק עצמו: מטרת המשחק, תיאור כללי של המשחק, מבנה המסמך כולו ורשימת מקורות עזר בכתיבת הפרויקט.

הפרק השני- פירוט התוכנה עצמה. תיאור מפורט של המשחק, ממשק אדם-מכונה, יכולות המערכת, תיאור האלגוריתם ורשימת נתונים.

הפרק השלישי- פירוט תכנון המערכת. תיאור המודולים השונים ומבנה הנתונים.

הפרק הרביעי- הוראות הפעלה. פירוט הוראות התקנה והפעלה.

והפרק האחרון – מכיל את כל קוד המקור של הפרויקט

**1.4 רשימה ביבליוגרפית:**

1) <http://stackoverflow.com/>

2) <http://www.java-forums.org> /

3) <http://www.java2s.com/>

**2. מפרטי תוכנה:**

יישום זה פותח בסביבת העבודה :



**2.1 תיאור כללי:**  
יש שני סוגי משחקים: שחקן נגד שחקן ושחקן נגד מחשב.

לאחר בחירת סוג המשחק ישנם הגדרות הבאות:

לשחקן נגד שחקן- בחירת מוד משחק, תיבת סימון אשר קובעת האם להפוך את הלוח בכל סיום תור מבחינה ויזואלית על-מנת להקל על כל שחקן בביצוע תורו.

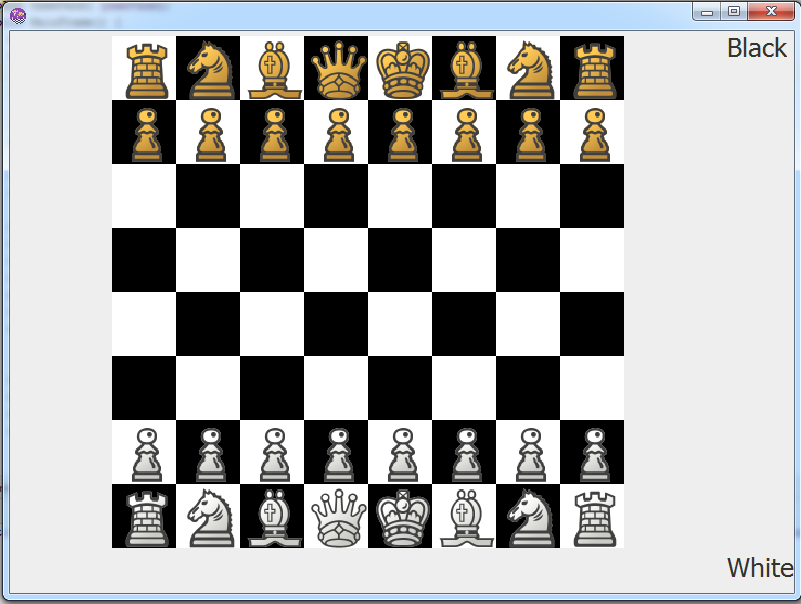
לשחקן נגד מחשב- בחירת מוד משחק, בחירת צבע השחקן, בחירת רמת הקושי(מ-1 עד 10).

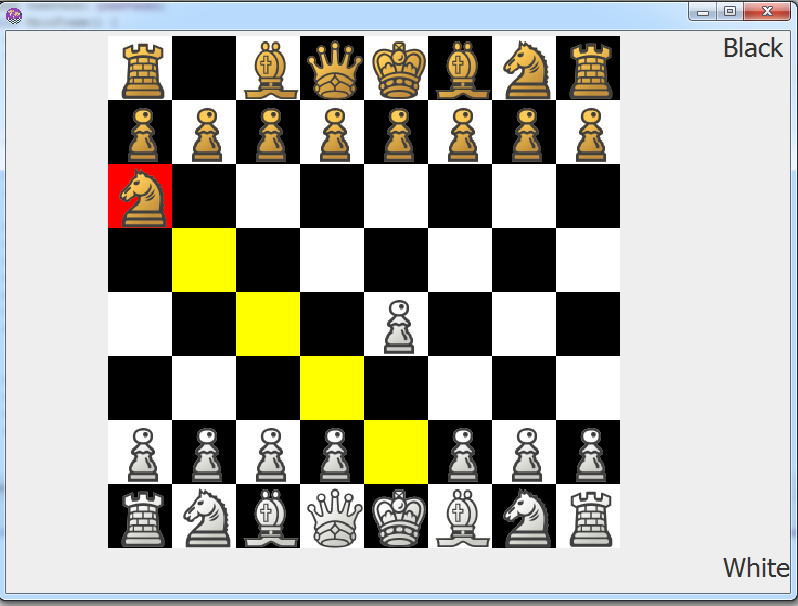
**2.1.2 מצב של סיום משחק:**

המשחק מסתיים כאשר יש שחמט, תיקו או סיום משחק אחר המוגדר בכל אחד מסוגי השחמט הניתנים לבחירה לפני המשחק.

**2.1.3 אסטרטגית המשחק:**

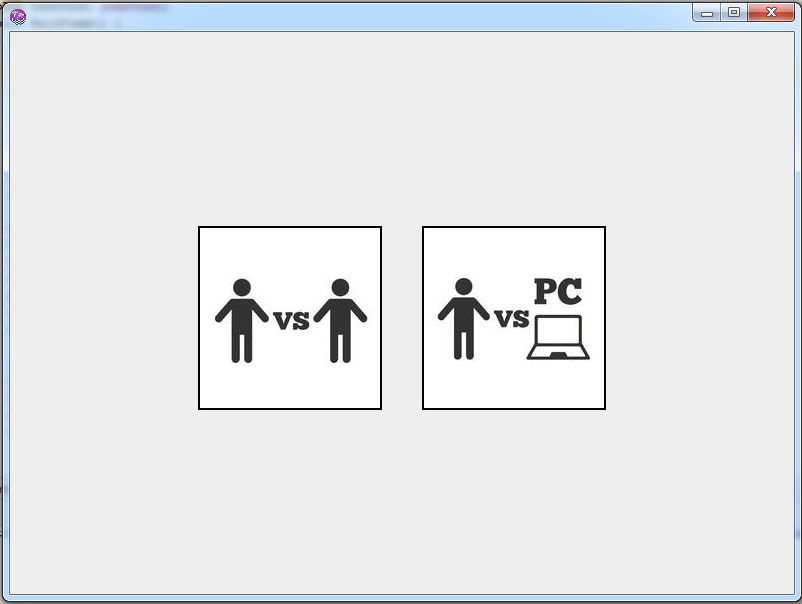
אסטרטגיית המשחק הינה לאכול כמה שיותר מכלי היריב ולבסוף לעשות שחמט או לנצח בדרך אחרת עפ"י סוג המשחק.



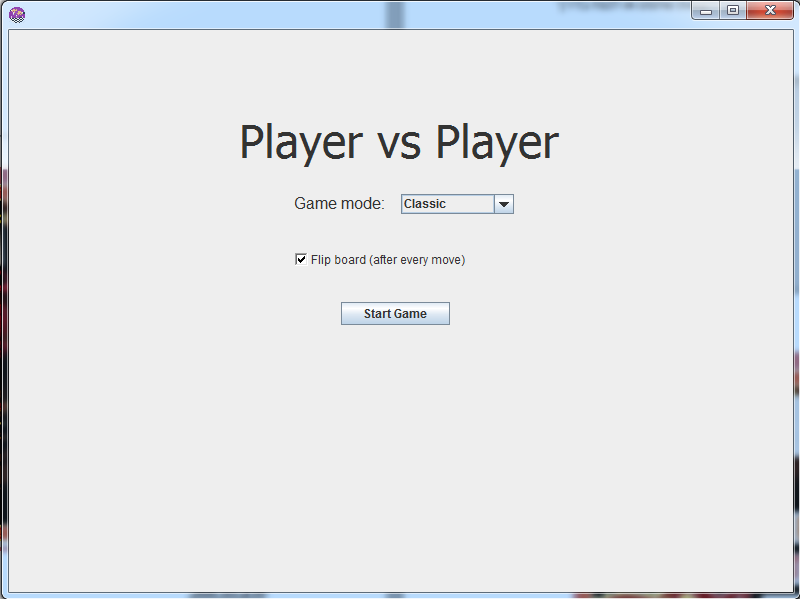


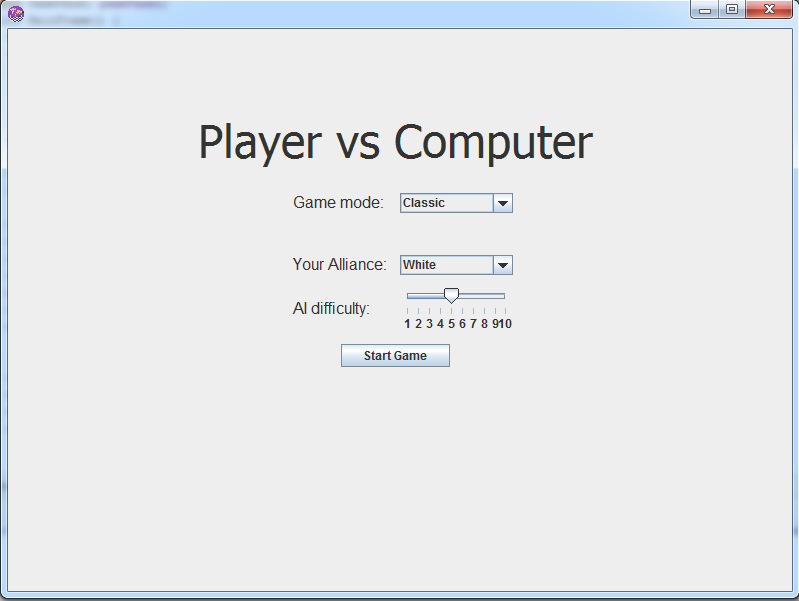
**2.2 ממשק אדם מכונה MMI:**

**תפריט פתיחה:**

 **לחיצה על:**  
**1) שחקן נגד שחקן -** תוביל למעבר לחלון ההגדרות לפני משחק של שחקן נגד שחקן.  
**2) שחקן נגד מחשב -** תוביל למעבר לחלון ההגדררת לפני משחק של שחקן נגד מחשב.

**מסך ההגדרות של שחקן נגד שחקן:**



**מסך ההגדרות של שחקן נגד מחשב:**  


* 1. **יכולות פונקציונאליות:**

1. המערכת מאפשרת משחק בין שחקן אנושי לשחקן מחשב ובין שחקן אנושי לשחקן אנושי.
2. המערכת דואגת שלא ניתן לבצע מהלכים לא חוקיים.
   1. **תיאור אלגוריתמים :**

האלגוריתם של שחקן המחשב יהיה מבוסס על **אלגוריתם מינימקס** **עם סינון אלפא-בטא**  
לפי אלגוריתם זה, המהלך הכדאי ביותר עבור המחשב הוא המהלך הכי פחות כדאי עבור השחקן האנושי. האלגוריתם יעבור על כל המצבים האפשריים כרגע בלוח, ידמה אותם, ינקד כל מהלך ולאחר מכן יוסיף את פרטי המהלך והניקוד לרשימה מקושרת.  
עומק הבדיקה בעצם מציין את מספר המהלכים קדימה שהמחשב מחשב, ככל שעומק הבדיקה גבוה יותר כך הסבירות להגיע למהלך האופטימלי עבור המחשב גבוהה יותר.  
בסיום הבדיקה מה שיתבצע זה המהלך שקיבל את הניקוד הגבוה ביותר.

סינון אלפא-בטא מאפשר מעבר יעיל יותר על עץ המינימקס ע"י דילוג על חיפושים מיותרים.

**סיבוכיות זמן האלגוריתם:**

Worst-case: O(b ^ d)

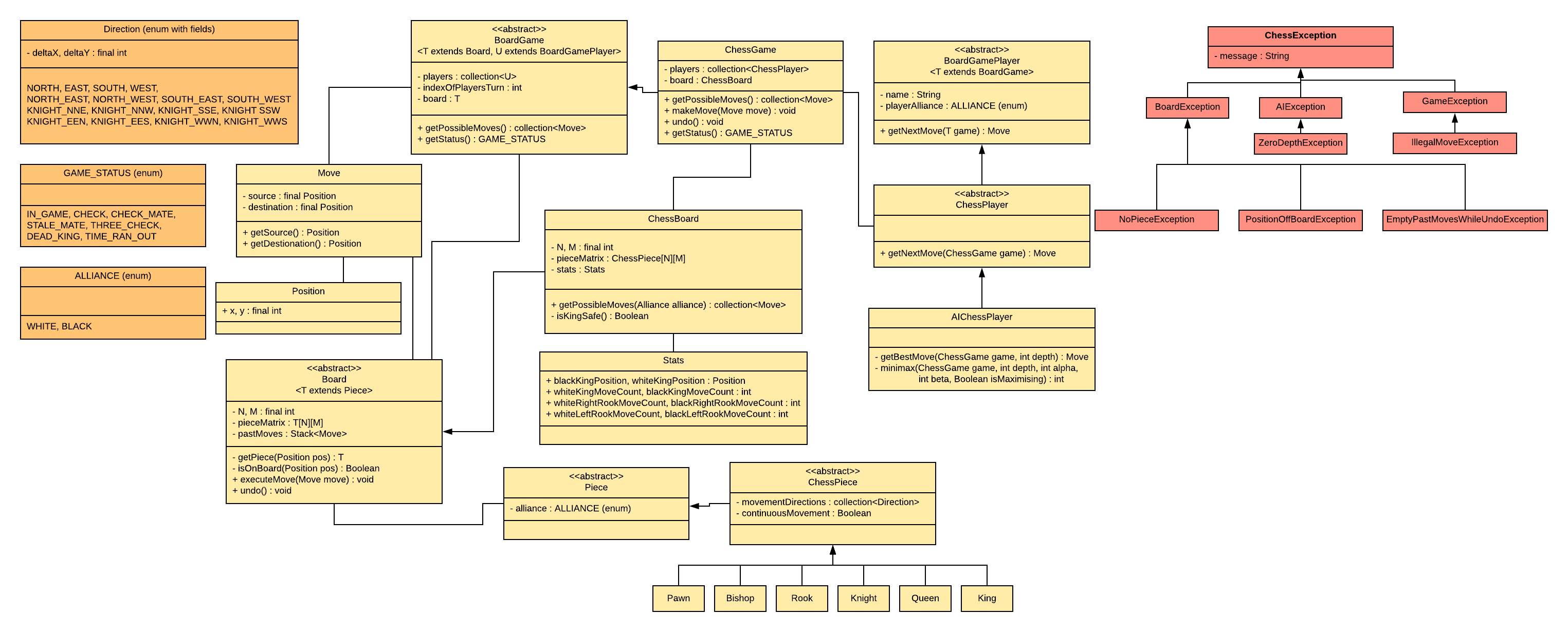
Average: O(sqrt(b) ^ d)

**פסודו קוד:**

|  |
| --- |
| getNextMove(game, depth): |
|  | maxValue = -∞ |
|  | for move in game.getPossibleMoves(): |
|  | game.makeMove(move) |
|  | value = minimax(game, depth - 1, -∞, +∞, false) |
|  | game.undo() |
|  | if value > maxValue: |
|  | maxValue = value |
|  | bestMove = move |
|  | return bestMove |
|  |  |
|  | minimax(game, depth, α, β, isMaximising): // minimax with alpha-beta puring |
|  | if depth = 0: |
|  | boardEvaluation = this.evaluate\_func(game.getBoard()) |
|  | return isMaximising ? boardEvaluation : -boardEvaluation |
|  | temp = isMaximising ? -∞ : +∞ |
|  | for move in game.getPossibleMoves(): |
|  | game.makeMove(move) |
|  | value = minimax(game, depth - 1, !ismaximising) |
|  | game.undo() |
|  | if isMaximizing: |
|  | if value > temp: |
|  | temp = value |
|  | α = max(value, α) |
|  | else: |
|  | if temp < value: |
|  | temp = value |
|  | β = min(value, β) |
|  | if α >= β: |
|  | break // stops the search for this tree because it's redundant |
|  | return temp |

**3. תכנון:**

**3.1 חלוקה למודלים ורשימת הנתונים:**

קישור לתמונה בגודל מלא:<https://raw.githubusercontent.com/OrtHolonJava/GilBenBasat/master/Chess_Uml.png>

**את הקוד ניתן למצוא בקישור הבא:** [**https://github.com/OrtHolonJava/GilBenBasat/tree/master/src**](https://github.com/OrtHolonJava/GilBenBasat/tree/master/src)

**מדריך למשתמש**

**4.1 הוראות התקנה**

**יש להיכנס לקישור הבא וללחוץ על GilBenBasat.jar:** [**https://github.com/OrtHolonJava/GilBenBasat/releases/tag/1.0**](https://github.com/OrtHolonJava/GilBenBasat/releases/tag/1.0)

לאחר ההורדה יש ללחוץ על הקובץ והוא יפתח.