מסמך אפיון דרישות

**משחק "חמקה"**

**פרטי הסטודנטים:**

* איה אבו אלהיג'א ת.ז: 207501370 [aya.15.667@gmail.com](mailto:aya.15.667@gmail.com)
* אמנה חסן ת.ז: 316021633 [amneh.hassan@gmail.com](mailto:amneh.hassan@gmail.com)
* נג'ואן כנג' ת.ז: 318714086 [n.kanj2@gmail.com](mailto:n.kanj2@gmail.com)
* ראוי ג'ראיסי ת.ז: 209146992 [rawige.98@gmail.com](mailto:rawige.98@gmail.com)

**A picture containing object, checker, table, food

Description automatically generated**

Table of Contents

[1. הקדמה 3](#_Toc54709909)

[1.1. מטרה 3](#_Toc54709910)

[1.2. Scope 3](#_Toc54709911)

[1.3. הגדרות 4](#_Toc54709912)

[1.4. סימוכין 4](#_Toc54709913)

[1.5. סקירה כללית של המשך המסמך 4](#_Toc54709914)

[2. תיאור כללי 4](#_Toc54709915)

[2.1. פרספקטיבת המוצר 4](#_Toc54709916)

[2.1.1. ממשק מערכת 5](#_Toc54709917)

[2.1.2. ממשק משתמש 6](#_Toc54709918)

[2.1.3. ממשק חומרה 6](#_Toc54709919)

[2.1.4. ממשק תוכנה 6](#_Toc54709920)

[2.1.5. משמק תקשורת 6](#_Toc54709921)

[2.1.6. אילוצי זיכרון 7](#_Toc54709922)

[2.1.7. פעולות (Operations) 7](#_Toc54709923)

[2.1.8. דרישות ספציפיות לאתר 7](#_Toc54709924)

[2.2. פונקציונליות המוצר 7](#_Toc54709925)

[2.3. אפיוני משתמש 7](#_Toc54709926)

[2.4. אילוצים 7](#_Toc54709927)

[2.4.1. הגבלות תוכנה וחומרה 7](#_Toc54709928)

[2.4.2. ממשק לתוכנות אחרות 8](#_Toc54709929)

[2.4.3. פעילות מקביליות 8](#_Toc54709930)

[2.4.4. דרישות אמינות 8](#_Toc54709931)

[2.4.5. אבטחה ובטיחות 8](#_Toc54709932)

[2.4.6. הקריטיות של היישום 8](#_Toc54709933)

[2.4.7. נהלים ומדיניות 8](#_Toc54709934)

[2.5. הנחות ותלויות 8](#_Toc54709935)

[2.6. הקצאת הדרישות 8](#_Toc54709936)

# הקדמה

## מטרה

מטרת המסמך הינה לאפיין דרישות עבור המשחק הממוחשב "חמקה".

המסמך מכיל אפיון דרישות שיענו על חוקי המשחק ואופן הפעלתו, ודרישות פונקציונליות לכל אובייקט.

המסמך מיועד לקריאה ע"י בעלי המוצר, צוות הפיתוח וצוות היישום.

## Scope

מסמך זה מאפיין את משחק הקופסא "חמקה" שהוא בעצם משחק "דמקה" בגרסתו הממוחשבת, ובהוספת כמה חוקים חדשים.

מטרת כל שחקן היא להוריד מהלוח ("לאכול") את כל החיילים של היריב, או לחסום אותם , כלומר לא לאפשר ליריב לקיים מהלך.

במהלך המשחק, השחקנים ייתקלו במגוון שאלות בנושאי הקורס "הנדסת תכנה". מענה נכון או לא נכון על שאלה יוכל להשפיע על מס' הנקודות שאסף במהלך המשחק. המשחק מיועד ל-2 משתתפים שהינם סטודנטים בקורס "הנדסת תוכנה ואבטחת איכות" ובסופו יש מנצח אחד; השחקו המנצח הוא השחקן שאסף יותר נקודות מהשחקן השני בסוף המשחק.

המשחק ישמש את הסטודנט במהלך הסמסטר ויהווה עזר נוסף ללמידה.

## הגדרות

Java: היא שפת תכנות מונחית עצמים.

Eclipse: פלטפורמת תוכנה פתוחה וחופשית לפיתוח יישומי צד – לקוח עשירים.

ממשק: ממשק תכנות יישומים – ספריות קוד, פקודות, פונקציות ופרוצדורות מן המוכן, בהן יכולים המתכנתים לעשות שימוש מבלי להידרש לכתוב אותן בעצמם.

אובייקט: מנגנון בתכנות מונחה עצמים המשמש לייצוג נתונים ופעולות בתוכנית מחשב.

מתודה: פרוצדורה או שיטה היא רצף של פקודות המאוגדות יחדיו, במטרה לבצע מטלה מוגדרת, מימוש של אלגוריתם. פונקציה היא שגרה המחזירה ערך.

שדה: אלמנט יסודי של אחסון נתונים.

## סימוכין

מפרט הדרישות נכתב בהתאם לתקן הנדר: STD – IEEE – SRS – 830 – 1998.

<http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>מקור:

מפרט הדרישות נכתב תחת: SRS template – organized by Classes/Objects

## סקירה כללית של המשך המסמך

**פרק 2 – תיאור כללי:** תיאור כללי של המערכת והממשקים בה. תיאור דרישות לא ספציפיות שינחו את פרק 3.

**פרק 3 – דרישות ספציפיות:** תיאור של הדרישות הפונקציונאליות מהמערכת, הכרת מבנה המערכת (מחלקות, שדות ופונקציות(.

# תיאור כללי

## פרספקטיבת המוצר

חמקה היא הדמיית משחק דמקה, מיועד ל- 2 שחקנים.

דמקה הינו משחק קופסא המורכב מלוח בגודל 8 \*8 של משבצות בשני צבעים לסירוגין (שחור ולבן) . כל שחקן מתחיל עם 12 חיילים בצבע כלשהו המסודרים על 3 השורות הקרובות אליו במשבצות השחורות, כאשר הלוח מונח כך שהמשבצת הימנית ביותר בשורה הקרובה לכל שחקן תהיה לבנה.

מטרת כל שחקן היא להוריד מהלוח ("לאכול") את כל החיילים של היריב, או לחסום אותם , כלומר לא לאפשר ליריב לקיים מהלך. שחקן שנשאר ללא חיילים או ללא מהלך אפשרי מוכרז כמפסיד.

כל שחקן מניע בתורו חייל-משחק באלכסון ממשבצת שחורה אחת למשבצת שחורה פנויה וסמוכה בכיוון היריב (תזוזה באלכסון). חייל רגיל לא יכול לבצע צעדים "אחורה" (לכיוון ממנו בא(.

"אכילת" חייל יריב מתבצעת כאשר חייל משחק של היריב מונח במשבצת סמוכה לחייל משחק של השחקן, ומעבר לחייל של היריב יש משבצת פנויה, כך שניתן לדלג מעל החייל של היריב ו" לאכול" אותו. חייל ש"נאכל" יוצא מהמשחק ומקנה נקודות ליריב. במידה וניתן לבצע פעולת " אכילה "– חובה לבצע אותה, במידה ולא בוצעה החייל שיכול היה לבצע אותה "נשרף" ויוצא מהמשחק. אם לאחר אכילת חייל כלשהו מתאפשרת אכילה נוספת ע"י אותו חייל, חובה לבצע גם אותה )באותו תור(. לא ניתן "לאכול" אחורנית, אולם ניתן "לאכול" אחורנית )ואף חובה לעשות זאת( כבמסגרת רצף של "אכילות".

כאשר חייל מגיע לשורה האחרונה ונעצר שם, הוא הופך ל"מלכה". מלכה בניגוד לחייל רגיל יכולה לנוע באלכסון לכל הכוונים (גם אחורה), ללא הגבלה על כמות המשבצות בדרך )מותר לבצע צעדים ארוכים( ועל כן "לאכול" חיילים/מלכות במרחק גדול.

בחמקה שעליכם לבנות למלכה יש יכולת מיוחדת נוספת והיא מעבר דרך קירות, לדוגמא, מלכה יכולה לעבור דרך הגבול הימני של לוח המשחק ולהיכנס חזרה ללוח דרך הגבול השמאלי של הלוח )במקום שהייתה מגיעה אליו אילו היינו מצמידים לוח נוסף מימין ללוח הקיים – הרעיון הוא לממש תנועה טבעתית(.

מלכה חייבת לבצע " אכילה" רק כאשר חייל יריב צמוד אליה, אך לא כאשר הוא נמצא בהמשך האלכסון )למרות ש"האכילה" אפשרית).

### ממשק מערכת

המערכת היא משחק מחשב מרובה שחקנים, המשחק הינו מקומי (כל השחקנים משחקים על אותו המחשב(.

במהלך כל המשחק יוכלו כל השחקנים לראות את מצבו של כל שחקן מבחינת נקודות, חיילים במשחק, וחיילים נאכלו).

פעולות כל משתמש מוצגים לשחקן השני. כאשר משתמש דורך על משבצת צהובה תופיע לו שאלה מסוימת שיידרש לענות עליה. מאגר השאלות והיסטוריות המשחקים ישמרו בקובץ Json.

### ממשק משתמש

בממשק המשתמש יהיה מסך ניווט שיכיל אופציות בחירה למשתמש ולחיצה על כל אחד מהם יוביל לפונקציונאליות אחרת במערכת- התחלת משחק, צפייה בהיסטוריות משחקים (משתמשים ותוצאותיהם במשחקים הקודמים), קריאת חוקי משחק וניהול מאגר שאלות (בו ניתן לצפות, לערוך, להוסיף ולמחוק שאלות).

במסך המשחק עצמו יצטרכו המשתתפים להזין את השם שלהם ואז יוצג לוח המשחק עצמו עם שמות השחקנים ומספר נקודות נוכחי לכל שחקן, זמן מתחילת המשחק וזמן מתחילת התור הנוכחי, שאלה והתשובות האפשריות שלה (רק בעת דריכה כל משבצת צהובה). המשבצות השונות יהיו בעיצובים שונים לפי סוגם.

כל משתמש בתורו יעשה צעד בחייל מבחירתו ממשבצת לשנייה לפי חוקי המשחק במטרה לחסום כמה שיותר אפשריות לשחקן השני במטרה לזכות.

כאשר משתמש עושה פעולה חוקית בתורו לפני תום זמן התור אז הוא מקבל נקודות זכות תלויות בזמן הנתר (יותר זנן נותר, יותר נקודות זכות), אבל כאשר עושה הפעולה אחרי שזמן התור הסתיים אז יורד ממנו נקודות (אפשר לקבל מס' נקודות שלילי).

דריכה על משבצת צהובה מציגה לשחקן שאלה עם תשובות אפשרויות, אם השחקן עונה נכון על השאלה אז הוא מקבל נקודות זכות (100 נק' לשאלה קלה, 250 נק' לשאלה בינונית, 500 נק' לשאלה קשה) ותסומן התשובה בירוק, אבל אם עונה בצורה לא נכונה יורד ממנו נקודות (250- נק' לשאלה קלה, 100- נק' לשאלה בינונית, 50- נק' לשאלה קשה), והתשובה הלא נכונה תסומן באדום. הממשק צריך להיות נוח, ידידותי למשתמש, מושך את העין ומותאם למסכי .PC

### ממשק חומרה

המערכת תתאים למחשב סטנדרטי (PC). המשחק יוצג על המסך ותפעולו יתבצע ע"י מקלדת ועכבר.

### ממשק תוכנה

המערכת תיכתב בשפת התכנות Java, הפיתוח יתבצע בסביבת Eclipse שהיא סביבה חינמית.

### משמק תקשורת

המערכת אינה מקוונת ולא תדרוש חיבור לאינטרנט מלבד הורדה חד פעמית בתחילת השימוש מאתר הקורס.

### אילוצי זיכרון

מאגר השאלות והיסטורית המשחקים שמורים בקבצי Json.

למערכת יוקצה זיכרון ב repository שלא יעלה על MB200.

לשם הפעלת המשחק יידרש חלק מזיכרון ה-RAM של המחשב במיקומי.

### פעולות (Operations)

|  |  |
| --- | --- |
| * התחלת משחק | * יציאה מהמערכת |
| * צפייה בהיסטורית משחקים | * צבירה / ירידת נקודות |
| * הזנת שם משתמש (כינוי) | * קריאת חוקי המשחק |
| * הזזת חיילים | * יציאה מהמשחק |
| * הצגת מאגר שאלות |  |
| * הוספת שאלה |  |
| * מחיקת שאלה |  |
| * עריכת שאלה |  |
| * מענה על שאלות רנדומליות בדרגות קושי שונות |  |
| * בדיקת תשובת השאלה |  |
| * טעינת מצב בדיקה לתחילת משחק |  |

### דרישות ספציפיות לאתר

לא נדרש, במשחק אינו מקוון.

## פונקציונליות המוצר

* קיום משחק "חמקה" בעל 2 משתתפים.
* חישוב ושמירת נקודות לכל משתמש.
* ניהול מאגר שאלות טריוויה בנושא קורס "הנדסת תוכנה".
* שמירת היסטורית המשחקים.

## אפיוני משתמש

המשתמשים העתידיים הינם סטודנטים בחוג "מערכות מידע", הלומדים את הקורס הנדסת תכנה ואבטחת איכות שזהו קורס מתקדם בתואר ולכן הם בעלי ידע רב בנושאי מערכות מידע ותפעול המשחק יהיה אינטואיטיבי בשבילם. השאלות עליהם הנשאלים צריכים להשיב הם מתוך תחומי הקורס, זאת על מנת לאפשר למשתמשים לבחון ולשפר את הידע שלהם בצורה משחקית ומהנה.

## אילוצים

### הגבלות תוכנה וחומרה

נדרש מחשב סטנדרטי עם מערכת הפעלה התומכת ב Java 8 לפחות, מסך, מקלדת, ועכבר.

### ממשק לתוכנות אחרות

לא קיימת התממשקות עם תוכנות אחרות.

### פעילות מקביליות

לא תאופשר פעילות מקבילה במשחק. כל שחקן ישחק בתורו ללא אפשרות לבצע מהלכים מקבילים.

### דרישות אמינות

המערכת תחשב את הוספת והורדת הנקודות לכל משתמש באופן מדויק לפי חוקי המשחק, המערכת תבחר שאלות באופן התואם את חוקי המשחק, מאגר השאלות ותשובותיהם יהיו נכונות ומדויקות, שמירת נתוני המשחקים הקודמים תהיה מדויקת, סדר השחקנים ייקבע רנדומלית ולאו דווקא בסדר כניסתם למשחק. הצגת מצב השחקנים תהיה מדויקת. בדיקת תקינות הצעדים שהשחקנים מבצעים.

### אבטחה ובטיחות

במערכת לא קיים מידע סודי או רגיש ולכן אין צורך לאבטח או להצפין את המידע.

### הקריטיות של היישום

ישנן מערכות רבות אשר מיישמות את משחק הקופסא הקלאסי "דמקה", התוספת במערכת שאנו נבנה תהיה הוספת השאלות בתחומי הקורס "הנדסת תכנה ואבטחת איכות" כחלק מהמשחק וכך לאפשר לסטודנטים של הקורס דרך קלה ומהנה לחזור על החומר, ללא תחושת למידה. המשחק פונה לאחוז אוכלוסייה קטן ולכן רמת הקריטיות הינה נמוכה.

### נהלים ומדיניות

עמידה תחת אילוצי הנהלים של אוניברסיטת חיפה בכלל וחוג "מערכות מידע" בפרט. שמירה על חוקי המשחק.

## הנחות ותלויות

* לא ניתן לשחק ולערוך את מאגר השאלות באותו הזמן.
* שני המשתמשים לא יכולים לענות על שאלה באותו הזמן.
* במשחק לא תתאפשר פעילות של שני שחקנים במקביל.
* לא ניתן לפתוח את חלון המשחק אם כבר פתוח.

## הקצאת הדרישות

ייתכנו דרישות נוספות בהמשך.

# דרישות ספציפיות

## דרישות ממשק חיצוני

### ממשק משתמש

* מסך הבית של המערכת יהיה מסך ניווט שיכיל את הפעולות הבאות: התחלת משחק, צפייה בהיסטוריית משחקים, חוקי המשחק וכלי ניהול למאגר השאלות. הכפתורים במסך זה יהיו מסודרים אחד מתחת לשני ויפתחו בלחיצת העכבר עליהם (עדיפות גבוהה-חלק מדרישות המטלה(.
* מסכי הניהול יכילו את הפעולות הבאות שיוצגו על גבי כפתורים: הוספת שאלות, עריכת שאלות, מחיקת שאלות, ויגיבו ללחיצות העכבר (עדיפות גבוהה-חלק מדרישות המטלה).
* הממשק יהיה ידידותי, פשוט, ונוח למשתמש (עדיפות גבוהה-יקל על השחקנים(.
* בעת לחיצה על כפתור "התחלת משחק" יפתח מסך לקליטת הכינוי לשני השחקנים.
* מסך ללוח המשחק שבו המצב ההתחלתי למשחק , ואפשרות ללחיצה על המשבצת האפשרית הרצויה
* לוח המשחק ייבנה ויתוכנן מראש כך שבמהלך המשחק הניקוד של כל שחקן יוצגו על גבי הלוח (עדיפות גבוהה-חלק מדרישות המטלה(.
* לאחר סיום תור של משתמש, יפתח חלון שיתריע על תורו של השחקן הבא (עדיפות גבוהה – בשחקנים צריכים לדעת התור של מי לשחק).

### ממשק חומרה

הפעלתו המלאה של המשחק תדרוש מחשב PC שכולל מסך, מקלדת ועכבר, משום שהמשחק מצריך לחיצה על כפתורים בנוסף להזנת פרטי כניסה לתחילת המשחק. בנוסף נצטרך את המקלדת עבור הזנת שאלות ותשובות לשאלות שיישאלו במהלך המשחק (עדיפות גבוהה-אביזרים נחוצים למשחק). אם המשחק יכלול צלילים נרצה שיהיו אביזרי שמע (כגון רמקולים) אך זו אינה דרישה בעדיפות גבוהה שתשפיע על מהלך המשחק או תפעולו.

### ממשק תוכנה

המערכת תיכתב בשפת התכנות Java התומכת במספר רב של מערכות הפעלה, ועל כן אין צורך לכתוב גרסאות שונות כדי שהמערכת תפעל במחשבים שונים בעלי מערכות הפעלה שונות. - הפיתוח יתבצע בסביבת Eclipse, שהיא סביבה חינמית. המערכת תפעל במערכת הפעלה תומכת Java. המערכת תיבדק במערכות ההפעלה Windows 8 & Windows 10 על מנת שתוכל לפעול במערכות אלה.

### משמק תקשורת

הורדת המערכת תיעשה דרך אתר הקורס ודפדפן האינטרנט במחשב השחקן ולשם כך נצטרך חיבור אינטרנטי. המערכת תעבוד על מחשב מקומי, אחד פר משחק, ולשם הפעלת המשחק לא יידרש חיבור אינטרנטי או רשתי.

## אובייקטים / מחלקות

### Game

מחלקה מייצגת משחק נוכחי.

**שדות:**

* Int id - מזהה משחק
* Player player1 - אובייקט שחקן ראשון
* Player player2 - אובייקט שחקן שני
* Date gameDate - תאריך תחילת משחק
* board - מערך בעל 64 משבצת ( Tile), המייצג לוח משחק
* Player winner - אובייקט שחקן ששומר בתוכו השחקן המנציח
* Time gameDuration - משך המשחק
* Player turnToPlay - אובייקט שחקן שאומר תור מי לשחק

**מתודות:**

* initGame - לצורך אתחול המשחק
* move - ביצוע השחקן להזזת חייל מסוים
* moveValidation - לצורך ווידוי שהמלך תקין
* upgradeQueen - הפיכת חייל למלכה
* checkAvailableMoves - הצעת המהלכים האפשריים למצב מסוים
* showSuggestedSquare - מתודה המקבלת צבע מסוים, מגרילה ומציגה משבצות על הלוח לפי תפקיד כל צבע
* pauseGame - השהיית המשחק
* resumeGame - המשכיות המשחק לאחר השהיה
* finishGame - גמירת המשחק
* popQuestion - להצגת שאלה רנדומלית ברגע הגעה למשבצת צהובה

**מסרים:**

* המשחק התחיל.
* המשחק הסתיים, המנצח הוא:\_\_\_\_\_, זמן: \_\_\_\_\_\_ .
* המשחק בהשהיה.
* המשחק ממשיך.
* תשובה נכונה ☺, צברת \_\_\_\_ נק'.
* תשובה לא נכונה ☹, הפסדת \_\_\_\_ נק'.

### Question

מחלקה מייצגת שאלה.

**שדות:**

* Id - מזהה שאלה
* text - הטקסט של השאלה
* answers - מבנה נתונים ששומר את התשובות האפשרויות
* rightAnswer - התשובה הנכונה
* difficulty - רמת קושי

**מתודות:**

* addAnswer - הוספת תשובה אפשרית
* removeAnswer -מחיקת תשובה אפשרית

**מסרים:** אין

### Player

מחלקה המייצגת שחקן במשחק.

**שדות:**

* username - כינוי שחקן
* Score - מספר נקודות צוברו

**מתודות:**

* updateScore - עדכון מספר הנקודות

**מסרים:** אין

### Tile

מחלקה המייצגת משבצת בלוח משחק.

**שדות:**

* value - ערך מספרי שמייצג חייל
* color - צבע משבצת

**מתודות:** אין

**מסרים:** אין

### Const

מחלקה לשמירת הפרמטרים הקבועים

**שדות:**

* maxGameDurations - משך זמן משחק מקסימלי
* cols - מספר עמודות
* rows - מספר שורות
* maxYellowSquars - מספר משבצות צהובות לתור
* maxTurnDuration - זמן השכיות מקסימלי לתור שחקן כך שצובר נקודות על מהלך.

timeDurationTillSuggestGreenSquare - זמן עובר עד הצעת צעד אפשרי שמקבל עליו

* timeDurationTillSeuggestOrangeSquare - זמן עובר עד שמציגים משבצות כתומות
* easyRight - מספר נקודות שמקבלים כל תשובה נכונה לשאלה קלה
* mediumRight - מספר נקודות שמקבלים כל תשובה נכונה לשאלה בינונית
* hardRight - מספר נקודות שמקבלים כל תשובה נכונה לשאלה קשה
* easyWrong - מספר נקודות שמפסידים כל תשובה לא נכונה לשאלה קלה
* mediumWrong - מספר נקודות שמפסידים כל תשובה לא נכונה לשאלה בינונית
* hardWrong - מספר נקודות שמפסידים כל תשובה לא נכונה לשאלה קשה
* pointsForGreenSquare - מספר נקודות שמקבלים בהגעה למשבצת ירוקה
* eatPoints - מספר נקודות לאכילת חייל
* soldreisForPlayer - מספר החיילים לכל שחקן.

### Difficulty

מחלקה מייצגת רמת קושי – Enum

**ערכים:**

* EASY
* MEDIUM
* HARD

### SysData

מחלקה מרכזית שתחזיק את נתוני היסטוריית המשחקים ונתוני השאלות.

שדות:• Games - מבנה נתונים ששומר את היסטוריית המשחקים.• Questions - מבנה נתונים ששומר את השאלות במשחק.מתודות:• Initiate - אתחול משחק. יצירת אובייקט מסוג GAME.• loadData - טעינת נתוני משחקים מקובץ.• writeData - כתיבת נתוני משחק לקובץ.• loadQuestions - טעינת השאלות מקובץ.• writeQuestions - כתיבת השאלות לקובץ.• addQuestion - הוספת שאלה חדשה למאגר.• removeQuestion - מחיקת שאלה קיימת מהמאגר.• addGame - הוספת משחק למבנה הנתונים של היסטוריית המשחקים.

• popQuestion – הגרלת שאלה .

מסרים:

* אובייקטים נטענו בהצלחה!
* אובייקטים נשמרו בהצלחה !
* שאלה עודכנה בהצלחה!
* קיימת תקלה בטעינה/כתיבת הנתונים.