

דוח מערכת – משחק Scorpion Minesweeper Iteration 2

1. תיאור כללי של המערכת

המערכת היא משחק Minesweeper קואופרטיבי לשני שחקנים, המבוסס על Java ו-Swing. לכל שחקן יש לוח נפרד, אך הם חולקים מספר חימר משותף ו-динיקוד משותף. בונוסף למקשיים רגילים קיימים תא שאלת (QUESTION) ו-הפתעה (SURPRISE) שימושיים עמוק ומשמעות.

2. אופן הרצה המערכת

2.1דרישות מקדיימות

אופן הפעלה: JDK גרסה 19 ומעלה.

סביבת פיתוח מומלצת: IDEA IDE (או כל IDE שתומך בפרויקט Maven).

הפרויקט נמצא בתיקייה: Mine-Sweeper-Scorpion.

2.2 הרצת המערכת מקובץ JAR

בגרסת ההגשה המצורפת מופיע קובץ JAR שטומן להפעלה:

Mine_Sweeper_Scorpion.jar

אופן הפעלה:

1. ודאו שמותקן Java 19 ומעלה

2. לחיצה Double-Click על הקובץ Mine_Sweeper_Scorpion.jar על התיקייה.

3. חלון המשחק נפתח מיד

הרצה דרך טרמינל / CMD (אם Double-Click לא עובד):

java -jar Mine_Sweeper_Scorpion.jar

לאחר שנייה-שתיים יפתח חלון המשחק.

2.3 פתיחת הפרויקט ב-IntelliJ והרצה

פותחים את File - Open - IntelliJ.

בוחרים את התיקייה: Mine-Sweeper-Scorpion.

מודדים אם הש-SDK של הפרויקט מוגדר ל-19 Java (או תואם למה רשום ב-pom.xml).

ממתינים ל-Maven שיסיים את הטעינה (Import Maven project).

נגישים לקובץ:

src/main/java/View/MainFrame.java

מricane את המתודה `main` בקובץ `MainFrame` (קליק ימני וaz ()Run 'MainFrame.main' MainFrame (>MainFrame.main (()

נפתחת חלונית Swing שהיא החלונית הראשית של המשחק.

לאחר ההרצה נפתח **מסך הראשי (Main Panel)** שמן המשמש מנוט למסכים האחרים.

3. קליטים למערכת

3.1 מסך פתיחה – Start Panel

מסך נפתח לאחר לחיצה על מסך Start Game מהמסך הראשי (Main Panel)

במסך התחילה המשתמש מזמן:

שמות שחקנים

Player 1 Name – שדה טקסט לשם השחקן הראשון.

Player 2 Name – שדה טקסט לשם השחקן השני.

אם אחד מהשדות נשאר ריק, המערכת יכולה להשתמש בשם ברירת מחדל ("Player 1", "Player 2")

רמת קושי (Difficulty)

בחירה באמצעות כפתורי רדיו (RadioButton):

Easy – לוח קטן יותר, מספר מוקשים נמוך יותר, מספר מוגבל של תאי Question/Surprise.

Medium – לוח גדול יותר, יותר מוקשים ותאים מיוחדים.

Hard – לוח גדול עוד יותר, מספר גובה של מוקשים, ושלוב יותר מתאגר של תאי Question/Surprise.

רמת הקושי נקבעת לפי ה-enum Difficulty בקוד:

Easy – לוח קטן יחסית בגודל 9x9, עם מספר מצומצם של תאים מיוחדים וחימם התחלתיים גבוהים. רמה זו מתאימה לשחקנים מתחילהים שרצו למדוד את מכנייקות המשחק בצורה רגועה וברורה.

Medium – לוח בינוני בגודל 13x13, עם יותר תאים מיוחדים ומתאגר גדול יותר בניהול סיכונים ובשיתופ פעולה בין השחקנים.

מיועדת לשחקנים שכבר מכירים את המשחק ורוצים העלה רמה טبيعית.

Hard – לוח גדול במיוחד בגודל 16x16, עם כמות משמעותית של אתגרים ותתי-מצבים לא צפויים

3.2 קלט בזמן המשחק (Game Panel)

במהלך המשחק משתמש הקצה מזין/מבצע:

קליק שמאלית על תא – ניסיון לחשוף תא בלוח המתאים (לשחקן 1 או 2).

קליק ימני על תא – סימון/ביטול של דגל על תא חשוד כموkish.

אישורים בחולנות דיאלוג

הפעלה של תא QUESTION/SURPRISE (חלון ConfirmDialog ששואל אם להפעל).

הודעות ניצחון/הפסד (OK לסגירה).

לחיצה על כפתורי שליטה:

Restart – התחלת משחק חדש באותה רמת קושי.

Exit – סגירת חלון המשחק.

חוק תורות: רק השחקן הפעיל יכול ללחוץ על הלוח שלו. לחיצה על לוח היריב נחסמת ולא מבצעת פעולה. מענה לשאלות: בתא QUESTION מופיע חלון שאלה, והשחקן בוחר תשובה מתוך אפשרויות (Radio Buttons) ולחוץ אישור.

סימון דגל (קליק ימני) לא מסיים תור – התור מתהפך רק אחרי מהלך חוקי של חשיפה. ב-Dialog QuestionDialog הבחירה נעשית באמצעות Radio Buttons + כפתור OK.

3.3 מסך ראשי – (Main Panel)Main Menu

במסך הראשי המשתמש לא מזין טקסט, אלא מבצע ניוט באמצעות לחצנים (Buttons) למסכים השונים במערכת:

מעבר למסך Start Panel לצורך הזנת שמות ובחירה רמת קושי. •

מעבר למסך היסטוריה המשחקים. •

מעבר למסך הסבר על כללי המשחק. •

מעבר למסך ניהול שאלות (מיועד למנהל/ADMIN). •

קלט במסך זה: לחיצה על אחד הcptors בלבד (input).

4. פלט ותוצאות – תרחישים שונים

להלן תיאור מה מצופה לקבל כתוצאה מהפעלת המערכת במצבים שונים:

4.1 פתיחת משחק חדש

המשתמש מזין שמות + רמת קושי במסך הפתיחה.

לוחץ על כפתור Start Game

נפתח מסך המשחק עם:

שני לוחות (אחד לכל שחזור).

תצוגת חיים משותפים (אייקוני לבבות/מספר).

תצוגת ניקוד משותף.

תצוגת מספר המוקשים שנוטרו לאיתור לכל שחזור.

.(Pause / Restart / Exit)

4.2 חשיפת תא בטוח (Safe Cell)

כאשר השחקן לוחץ קליק שמאלית על תא ללא מוקש:

אם התא ריק עם 0 מוקשים סבירו (EMPTY) – המערכת מבצעת חשיפה וקורסיבית של אזור ריק (Cascade), כולל מספרים בשולטים.

אם התא מכיל NUMBER (יש מוקשים מסויב) – מוצג מספר המוקשים הסמוכים.

אם התא הוא QUESTION / SURPRISE – התא נחשף בסוג מיוחד, וניתן לבחור אם להפעיל אותו בחלון נפרד.

בכל חשיפת תא בטוח (לפי בדיקות ה-*Acceptance*):

הניקוד המשותף גדל ב-1 נקודה.
לאחר מהלך חוקי, התור מתחלף אוטומטית לשחקן השני.

4.3 סימון דגלים (Flags)

כאשר השחקן לוחץ קליק ימני על תא:

אם התא עדין לא נחשף – המערכת מסמנת/מסירה דגל.

כללי הניקוד (לפי *GameLogicAcceptanceTest*):

סימון דגל על מוקש אמיתי + 1 נקודה.

סימון דגל על תא שאינו מוקש - 3 נקודות (עונש).

תאי דגל אינם נחשפים כל עוד הם מסומנים – יש להסיר את הדגל כדי לחשוף אותם.

4.4 פגעה במקש (Mine)

כאשר השחקן חושף תא המוגדר כ-MINE:

מספר החיims המשותפים יורד ב-1.

התא מוצג כמקש (לפי הגרפיקה שנבחרה בלוח).

הניקוד אינו משתנה על חשיפת מוקש.

אם כמות החיims מגיעה ל-0 מצב המשחק משתנה ל-LOST.GameState.

מתבצעת חשיפה אוטומטית של כל התאים (הציגת כל המוקשים).

מצגת הודעה על הפסד כולל הניקוד הסופי.

4.5 הפעלת תא / Surprise

לפני הפעלה קופץ ConfirmDialog (כן/לא) להפעלת תא מיוחד.

אם אין מספיק ניקוד לא מתבצעת .activationCost

לאחר חשיפה, ניתן להפעיל תא מיוחד (אם יש מספיק נקודות):

הפעלה גבוהה מהשחקנים עלות בניקוד activationCost (לפי הרמה).

עבור תא SURPRISE – מתבצעת פעולה אקראית (Good / Bad):

הפתעה טובה: הוספה נקודות (לפי surpriseValue) + ניסיון להוסיף חיים.

הפתעה רעה: הפחיתה נקודות / חיים (לפי ההגדרות בקוד).

עבור תא QUESTION – התא מסמן שהפעלה שאלת (לוגיקה קיימת בקוד לשילוב שאלות).
תא QUESTION/SURPRISE ניתן להפעלה עם אחת בלבד: לאחר הפעלה התא מסומן כ-pseen, ולא ניתן להפעיל אותו שוב.

4.6 ניצחון במשחק

המשחק מסתיים באחד מהמצבים הבאים:

- **ניצחון (WIN):**

כאשר אחד משני השחקנים מגלה את כל המוקשים בלוח שלו.

- **הפסד (LOSS):**

כאשר כל החיימ המשותפים של השחקנים נגמרים.

עם סיום המשחק, מתריצעות הפעולות הבאות באופן אוטומטי:

- כל התאים בשני הלוחות נחשפים.
- המשחק עובר למצב סיום, ולא ניתן לבצע פעולות נוספות בלוחות.
- כל החיימ שנותרו לשחקנים מומרים לניקוד סופי.
- כל לב מומר למספר נקודות השווה למחיר ההפעלה של תא QUESTION או SURPRISE ברמת הקושי הנוכחית.
- מוצגת הודעה סיום משחק (ニッショウ או הפסד) הכוללת את התוצאה והניקוד הסופי.

בסיום משחק (WON/LOST) המערכת קוראת ל-Controller כדי לטע את המשחק להיסטוריה (players,) Game History (difficulty, score, lives, accuracy, duration

Restart / Exit 4.7

:Restart

המערכת יוצרת משחק חדש באותה רמת קושי.

מאפסת את הלוחות, הניקוד, החיימ והכללים.

מעדכנת מחדש את ממשך המשחק (לבבות, ניקוד, לוחות).

:Exit

סגורת את חלון המשחק וסיום הריצה.

5. מה יכול משתמש הקצה (השחקן) לעשות במערכת

5.1 מסכים עיקריים

מסך פתיחה (StartPanel)

הזנת שמות שני השחקנים.

בחירה רמת קושי (Easy / Medium / Hard).

התחלת המשחק בלחיצה על כפטור Start.

מסך משחק (GamePanel)

תצוגת שני הלוחות (BoardPanel אחד לכל שחקן).

הציג שמות השחקנים במסגרת מועצת.

הציג מספר חיים, ניקוד, מספר המוקשים שנותרו.

כפתורי Pause / Restart / Exit

דיאלוגים / חלונות קופצים

. QUESTION/SURPRISE אישור הפעלת תא

הודעת ניצחון / הפסד עם תוצאה סופית.

מסך היסטוריית משחקים (GameHistoryFrame): צפיה ברשומות של משחקים קודמים (שחקנים, תאריך/שעה, קושי, תוצאה, ניקוד, חיים, דיווק בשאלות, מושך).

מסך ניהול שאלות (QuestionManagementFrame): ניהול מאגר השאלות שנטען מה-CSV.

מסך ראשי (Main Panel / Main Menu)

מוך תפריט ראשי המשמש כניסה לכל המערכת. מהמוך ניתן לנוט למסכים:
Start Game, View History, How to Play, Manage Questions

5.2 פעולות המשתמש

לבחור לאיזה מסך הוא רוצה להיכנס: משחק, היסטוריה, ניהול שאלות.

לכתוב שמות ולבחר רמת קושי.

לחשוף תאים בלוח שלו.

לסמן דגלים על תאים חשודים.

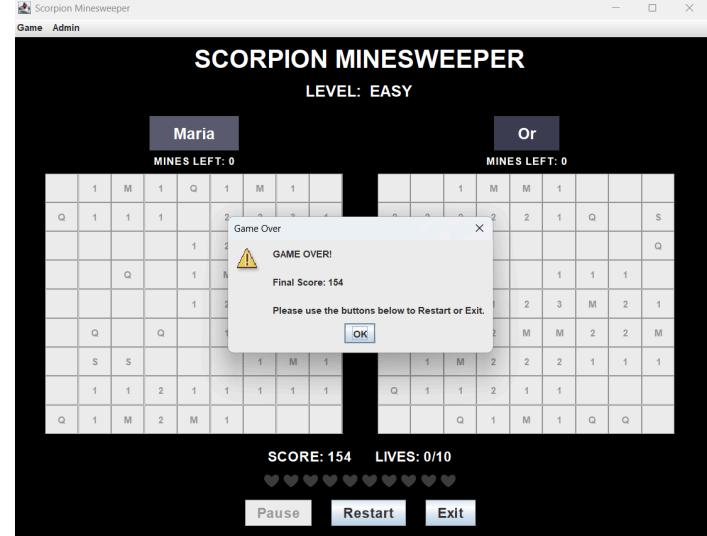
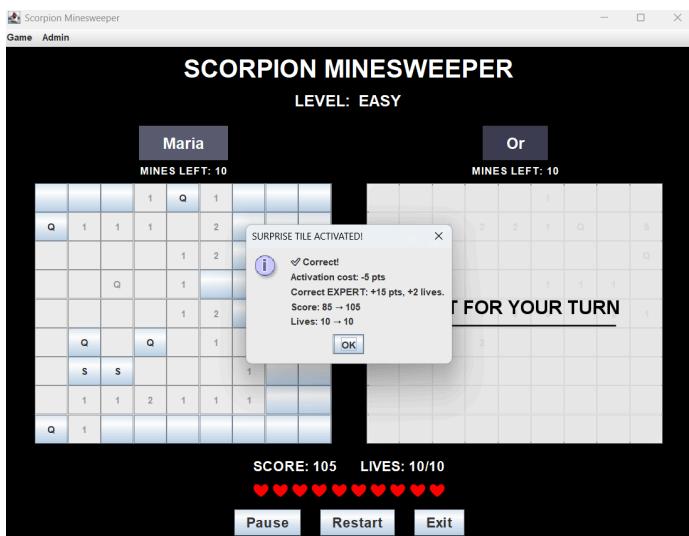
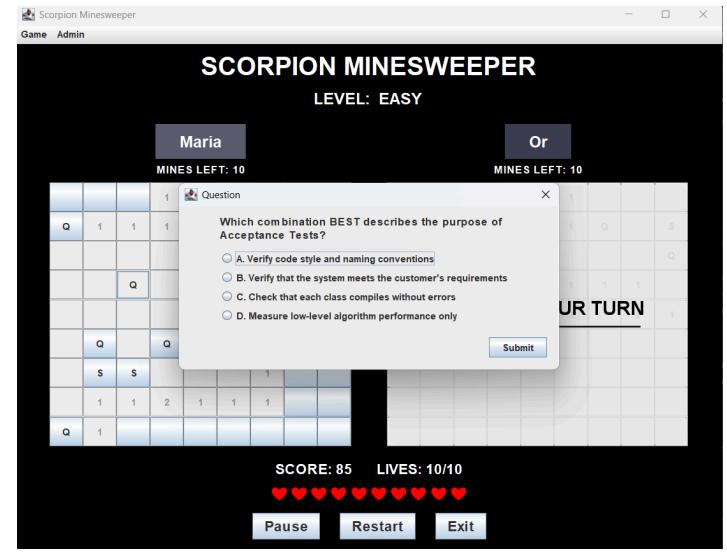
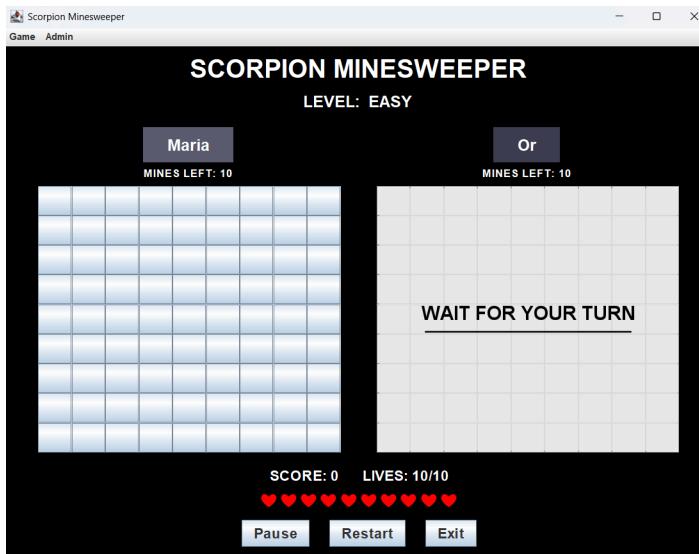
להחליט אם להפעיל תאים מיוחדים (שאלה/הפעעה).

לענות על שאלות

לצפות בהיסטוריה של משחקים.

לצפות בחלונות שמסבירו איך לשחק.

מנהל יוכל להיכנס למסך ניהול שאלות עם סיסמה "ADMIN".
לשימוש בפתרונות Exit / Pause / Restart



Game & Players History

Difficulty: All **Result:** All **Search:**

Games History							
Players	Date / Time	Difficulty	Final Score	Remaining Lives	Correct Answers	Accuracy	Duration
Maria + Or	12/12/25 21:51	EASY	69	0	0/0	-	0:37
Or + Alla	12/12/25 21:54	EASY	73	0	0/2	0%	0:138
ds + sd	16/12/25 20:27	EASY	94	0	0/3	0%	0:11:10
zxcv + zxcv	16/12/25 20:21	EASY	162	7	0/1	0%	0:25
scdfv + scdfv	18/12/25 20:28	EASY	106	0	0/0	-	0:118
Maria + Or	20/12/25 16:15	EASY	154	0	1/1	100%	0:37

Players History				
Player	Total Games	Best Score	Average Accuracy	Preferred Difficulty
ad	1	84	0%	EASY
Or	3	154	33%	EASY
zxc	1	162	0%	EASY
scdfv	1	106	0%	EASY
zxcv	1	162	0%	EASY
Maria	2	154	50%	EASY
ds	1	84	0%	EASY
Alla	1	73	0%	EASY

Question Management

ID	Text	A	B	C	D	Correct	Difficulty
2	In Agile development, what does the acronym 'TDD' stand for?	A detailed technical document.	A list of all bugs identified.	A short description of the system's requirements.	A meeting agenda.	2	EASY
3	What does the acronym 'BDD' stand for?	Software Requirements Specification.	System Release.	System Reliability.	2	EASY	
4	Which artifact is used to describe system requirements?	Unit test cases.	Class diagrams.	User manual.	Software Requirements Specification.	4	EASY
5	Which of the following is a type of testing?	Testing based on user stories.	Testing that focuses on specific requirements.	Testing done online.	Testing that checks for defects.	2	EASY
6	What is the goal of unit testing?	To rewrite the code.	To check coding standards.	To find defects in the code.	To measure the complexity of the code.	3	EASY
7	Which of the following is a type of testing?	It ignores the interface.	It is based on user stories.	It uses knowledge of the system's requirements.	It is always done.	3	MEDIUM
8	In Scrum, who is responsible for creating the product backlog?	Stakeholders.	Scrum Master.	Development Team.	Product Owner.	4	MEDIUM
9	What is the purpose of a 'Walkthrough'?	Inspection.	Pair programming.	Informal desk checks.	2	MEDIUM	
10	What is the main purpose of 'Test Driven Development'?	To describe how to model classes.	To specify boundary conditions.	To show the project.	2	MEDIUM	
11	What is the purpose of a 'JUnit' test?	A unit test.	A system test.	An acceptance test.	A usability test.	1	MEDIUM
12	In the context of software development, what does 'Mocking' refer to?	A simple implementation.	A tool that monitors the system.	A test code that interacts with the system.	A database used for testing.	3	MEDIUM
13	What is the purpose of 'Equivalence Partitioning'?	Boundary Value Analysis.	State Transition.	Random Testing.	2	HARD	
14	In Agile, which stage involves iterative development?	Iterations happen.	Iterations are long.	Each iteration results in a working product.	Testing is postponed.	3	MEDIUM
15	What is the purpose of 'Merge Conflicts'?	Defects are found.	Merge conflicts arise.	It removes the need for manual testing.	Build and test phases.	3	HARD
16	During an inspection, what role does the 'Recorder' play?	Reader.	Author.	Moderator.	Recorder.	3	MEDIUM
17	What is the main purpose of 'Test Automation'?	To optimize the test process.	To define how to write automated tests.	To write detailed test cases.	To estimate the time required for testing.	2	HARD
18	What is the metric 'High Percentage of Defects'? (in English)	High number of defects.	High percentage of defects.	Low execution time.	2	HARD	
19	You have a function that takes a list of integers and returns the sum of all even numbers. What is the name of this function?	Random Testing.	Equivalence Partitioning.	State Transition.	Ad-hoc Testing.	2	EXPERT
20	A team maintains a shared codebase. What is the best practice for committing changes?	Commit everything.	Use feature branches.	Use only one shared repository.	Push code to a central repository.	2	EXPERT

Add Edit Delete Save

6. בדיקות המערכת

בדיקות המערכת בוצעו בשילוב לוגיקה בקוד ובדיקות שימוש יומיומיים במסמך הגרפי.

6.1 בדיקות לוגיקה (מחלקות בדיקה בקוד)

בפרויקט קיימות מספר מחלקות בדיקה תחת test/src/test:

GameStartTest

בודק יצירת משחק חדש בכל רמת קושי.

מאמת את גודל הלוחות, מספר המוקשים, מספר תאי Question/Surprise ומספר החימום ההתחלתיתם.

בודק את תפקוד restartGame() ואת איפוס הנתונים.

RandomPlacementTest

يוצר מאות/אלפי לוחות אקרים לכל רמת קושי.

בודק שbulletן כולל מספר המוקשים, תאי השאלה ותאי ההפתעה תואם להגדרות difficulty.

מבצע בדיקה סטטיסטית שההתפלגות אקראית ונמצאת בטולרנס סביר.

MinesweeperLogicTest

בודק את הלוגיקה של חשיפת תאים:

.(Cascade) חשיפת שרשרת תאים ריקים

הציג מספרים נכונה סיבב מוקשים.

התנהגות גבולות הלוח והימנעות מגישה מחוץ לטווח.

GameLogicAcceptanceTest

בודק את חוקי המשחק לפי ה-SRS:

חסיפה תא בטוח → + 1 נקודה.

סימון מוקשים עם דגלים → + 1 נקודה למקש נכון, -3 נקודות לתא בטוח.

פגיעה במקש → הפחתת חיים, ללא שינוי ניקוד.

תנאי ניצחון והפסד, כולל סיום משחק והמרת חיים לניקוד.

UsedCellTest

מבטיח שלא ניתן "לנצל" אותה משבצת פעמים.

בודק שחשיפה חוזרת/סימון חוזר של אותו תא לא מօסיפה ניקוד נוסף ואינה משנה חייםשוב.

את מחלקות הבדיקה הללו ניתן להריץ דרך ה-IDE Java Applications (בכלן קיימת מתודת `main` שמדפיסה תוצאות לטובת בדיקה ידנית).

6.2 בדיקות ידניות (GUI)

בנוסף למחלקות הבדיקה בקוד, בוצעו גם בדיקות ידניות בפועל:

פותיחה משחק בכל רמת קושי והבטחה שהלוחות והחימם תואימים לrama.

. ניסיון לחסוף תאים שונים: מוקשים, תאים ריקים, תא NUMBER, תא Question/Surprise.

בדיקות התנהגות הדגלים:

סימון מוקשים אמיתיים וקבלת ניקוד מתאים.

סימון תאים בטוחים וירידת נקודות.

. בדיקת תנאי הפסד – חסיפה מוקשים עד סיום החיים והופעת הודעת Game Over.

בדיקת תנאי ניצחון – זיהוי כל המוקשים בלוח אחד (באמצעות סימון בדgel או חסיפה), והמעבר למצב WIN.

בדיקה כפטור Restart (שהכל מתאפשר בצורה תקינה) ו-Exit (סגירת התוכנה).

6.3 בדיקות Black Box (בדיקות פונקציונליות)

בדיקות Black Box בוחנות את התנהגות המערכת מנקודת מבט של משתמש קצה, ללא תלות במימוש הפנימי של הקוד.

בדיקות מתמקדות בתרחישי שימוש מלאים ובתגובה המערכת לקלטים שונים.

תרחישים שנבדקו:

- **בדיקה תנאי ניצחון (TC-BB-WONGAME-001):**
נבדקה התנהגות המערכת כאשר שחזור מזחה את כל המוקשים בלוח שלו. המערכת מציגה הודעת ניצחון, מחשבת ניקוד סופי, חושפת ומcpfיאה את שני הלוחות.
- **בדיקה ניהול תורות(001-TC-BB-TURN-001):**
נבדק כי רק השחקן הפעיל רשאי לפעול בלוח שלו.
לחיצה על לוח היריב נחסמת ואינה גורמת לשינוי במצב התא, בניקוד, בחימם או בתור.
- **בדיקה תאי QUESTION (TC-BB-QUESTION-001):**
נבדק כי בעת חשיפת תא מסווג Question מוצג תחילה חלון אישור (ConfirmDialog) להפעלת התא.
בעת אישור הפעלה מוצג חלון שאלה, ותשובה נכונה גורמת לתגמול מתאים בהתאם לחוק.
המשחק, תוך המשחק המשך במצב RUNNING.
- **בדיקה דגליים (TC-BB-FLAG-001):**
בדיקות אלו כיסו מחלוקת שקיים של מצבים חוקיים ולא חוקיים, והבטיחו תגובה תקינה של המערכת בכל אחד מהם.

בדיקות אלו כיסו מחלוקת שקיים של מצבים חוקיים ולא חוקיים, והבטיחו תגובה תקינה של המערכת בכל אחד מהם.

6.4 בדיקות White Box (בדיקות לוגיקה פנימית)

בדיקות White Box בוצעו לצורך בדיקת הזרימה הפנימית של הקוד ונתיבי הביצוע המרכזיים במערכת. בדיקות אלו התמקדו בענפי החלטה, תנאים פנימיים וمبرרי מצב, ולא רק בפלט החיצוני למשתמש.

תחומי בדיקה עיקריים:

- **чисוב תגמולים(TC-WB-SCORES-001):**
נבדקה הזרימה הפנימית של מגנון חישוב ניקוד וחימם בעקבות מענה על שאלות, בהתאם לרמת הקושי ולרמת השאלה.
- **בדיקות ידאו שהתגמולים מחושבים נכון ושמצב המשחק נשאר RUNNING כאשר אין תנאי סיום.**
- **ניהול תורות פנימי(TC-WB-TURNS-001):**
נבדקה הלוגיקה המודדת טור מעבר תוך מתבצע רק לאחר פעולה חוקית, וכי פעולות לא חוקיות אינן גורמות לשינוי מצב.
- **בחירה שאלה אקראית(TC-WB-QUESTION-001):**
נבדקה הלוגיקה לבחירת שאלה מתוך מאגר שאלות טען, כולל ידוא שהמאגר אינו ריק ושלא מוחזרת שאלה לא חוקית.

- **פעולת דגל פנימית(001-TC-WB-FLAG):**
נבדקה הזרימה בה פעולה סימון דגל אינה מסיימת את תור השחקן, כל עוד מדובר בתא חוקי לפעולה.

בדיקות אלו הבטיחו שכלי ענפי החלטה המרכזיים בקובד מתבצעים כמצופה, גם במצבים שאיןם גלויים ישירות למשתמש.

6.5 בדיקות אוטומטיות (Unit Test)

(1) בדיקות מערכת התורות (TurnSystemTest)

נבדקה התנהוגות המערכת במצבים של:

- ניסיון של שחזור פעיל ללחוץ על לוח היריב (הפעולה נחסמת/מוזנחת, ללא שינוי ניקוד/חימום/טור).
- ביצוע מהלך חוקי בלוח של השחקן הפעיל - חשיפת תא ומעבר תור אוטומטי לשחקן השני.
- תרחיש זרימה משולב: שחזור 1 מהלך חוקי - מנסה שוב (נחסם) - שחזור 2 מבצע מהלך חוקי - התור חוזר לשחקן 1.

בדיקות אלו משלבות בדיקות Box ו-White Box ומודאות זרימה תקינה של מהלך המשחק.

(2) בדיקות מנגן דגליים (FlagDoesNotEndTurnTest)

נכתבה בדיקת Unit המאמת כי פעולה של סימון/הסרת דגל (קליק ימני) על תא נסתר אינה מסיימת את תור השחקן, כלומר ה-callback נקרא עם endedTurn = false.

(3) בדיקות מערכת השאלות – בחירת שאלה רנדומלית (QuestionManagerTest)

בוצעו בדיקות אוטומטיות עבור QuestionManager.getRandomQuestion() () במטרה לוודא:

- מוחזרת שאלה שאינה תלlich כאשר קיימות שאלות במאגר.
 - השאלה המוחזרת קיימת בראשימת השאלות של המנהל.
 - לא נזורך חריגות כאשר הרשימה אינה ריקה.
 - בוצעו קריאות רבות לוודא עקבויות: תמיד מוחזרת שאלה חוקית מתוך המאגר (Random).
- תקין אך לא יוצאה מגבולות הנתונים).

(4) בדיקות תגמול לשאלות – תשובה נכונה לפי טבלת ניקוד/חימום (QuestionCorrectAnswerTest)

בוצעו בדיקות לוגיקה (לא או) לפונקציה `processQuestionAnswer(...)` עבור מספר קומבינציות של:

- רמת משחק (EASY / MEDIUM / HARD)
- רמת שאלה (EASY / MEDIUM / HARD / EXPERT)
הבדיקות מודאות:
 - ה尼克וד מתעדכן בהתאם לכלי התגמול.
 - מספר החיים לא יורד (ובחלה מהמרקם עולה).
 - מצב המשחק נשר ARUNNING.
נבדקו גם מקרי קצה כגון:
 - צבירת ניקוד במספר תשובה נכונות רצופות.
 - מצב של הגעה במספר חיים מסוים והמרת "חימ_us" לניקוד (לא שינוי לרעה במצב המשחק).

5) בדיקות תא שאלה "ננעל" לאחר שימוש (QuestionCellUsedAfterAnswerTest)

- נכתבו בדיקות המאמנות שתא מסוג NOITEM מסומן כ-used לאחר הפעלה (גם בתשובה נכונה וגם בתשובה שגוייה), ולכן:
- הפעלה הראשונה מצילה ומסמנת את התא used.
 - ניסיון הפעלה חוזר של אותו תא נכשל (לא ניתן להפעיל שוב).
 - בדיקה שתאים שונים מסוג QUESTION עובדים עצמאית (כל תא ננעל רק אחרי שהופעל).
 - יציבות משחק: ניסיון להפעיל תא שכבר used לא משנה ניקוד/חימ, לא מפיל את המשחק, ומצב המשחק נשר ARUNNING.

6) בדיקת זרימת מערכת השאלות (QuestionSystemTest)

- נכתבה בדיקה בסגנון Black Box לזרימה מלאה:
- סימולציה של חשיפת תא QUESTION במהלך משחק רץ.
 - תשלום "עלות הפעלה" (activation cost) לפי רמת קושי המשחק.

- סימולציה של תשובה נכונה - תגמול נקודות וחימם לפי הכללים.
הבדיקה מאמתת חישוב ניקוד נכון (כולל עלות הפעלה + נקודות חשיפה + תגמול), עדכו
חימם, ושמירה על STATE.RUNNING.Game.

7) בדיקות חישוב תגמול לשאלות (ScoreCalculationTest)

נכתבת בדיקה המאמת עדכון ניקוד וחימם עבור תשובה נכונה לשאלת ברמת EXPERT (במשחק EASY), תוך שמירה על מצב משחק RUNNING.

8) בדיקות תנאי ניצחון (WinGameTest)

נכתבת בדיקת תנאי ניצחון המדמה סימון כל המוקשים בלוח באמצעות דגליים.
הבדיקה מאמתת:

- מעבר מצב המשחק ל-WON.State.Game
- עצירת המשחק (isGameRunning=false, isGameOver=true).
- חישוב ניקוד סופי חיובי.
- חשיפה ונעילה אוטומטית של הלוחות כך שלא ניתן לבצע פעולות נוספות לאחר הניצחון (כולל ניסיון reveal שמחזיר false).

הרצה בדיקות אוטומטיות

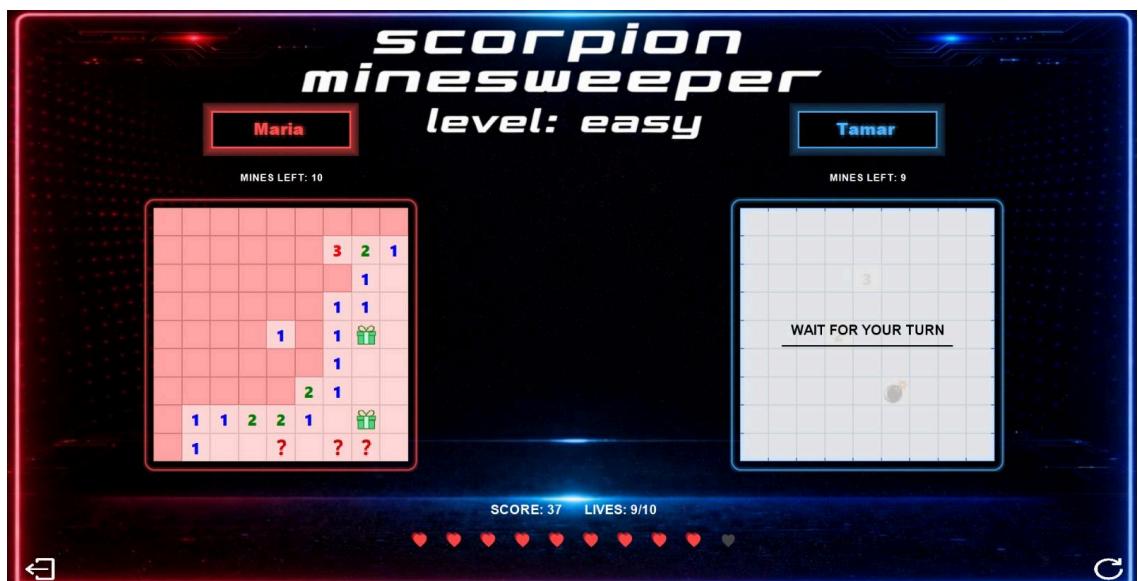
בדיקות האוטומטיות בפרויקט מוצאות באמצעות סביבת הפיתוח IDEA IntelliJ. Test Sources Root, ולכן ניתן להריץ את הבדיקות ישירות מתוך ה-IDE.
אופן ההרצה:

- הרצת בדיקה בודדת: קליק ימני על מחלקת בדיקה - Run.
- הרצת כל הבדיקות: קליק ימני על תקיית הבדיקות - Run Tests.

6.6 סיכום הבדיקות

שילוב בדיקות ידניות, בדיקות Black Box, White Box ובדיקות אוטומטיות סיפק כיסוי רחב של התנהלות המערכת.
בדיקות מאשרות שהמשחק פועל בצורה יציבה, עומדת בדרישות שהוגדרו, ומטפל כראוי במצבים קצה ותרחישים מורכבים.

*תיקון תמונות מעודכנות מאייטרציה 2





question management

Difficulty: All | Correct: All | ID: Apply | Clear

ID	Text	A	B	C	D	Correct	Difficulty
2	In Agile development, what... A detailed technical design.	A list of all bugs in the proj.	A short description of functi...	A meeting agenda for the s...	3	EASY	
3	What does the acronym S... Software Review Specifica...	Software Requirements Sp...	System Release Schedule	System Reliability Standard	2	EASY	
4	Which artifact is typically pr... Unit test cases	Class diagrams	User manual	Software Requirements Sp...	4	EASY	
5	Which of the following BE... Testing based on the inter...	Testing that focuses on inp...	Testing done only by the d...	Testing that checks perfor...	2	EASY	
6	What is the goal of a Code... To rewrite the code from s...	To check coding style only	To find defects and improv...	To measure the execution ...	3	EASY	
7	Which of the following is T... It ignores the internal struct...	It is based on the GUI only	If it uses knowledge of the pr...	It is always done by the cu...	3	MEDIUM	
8	In Scrum, who is mainly res... Stakeholders	Scrum Master	Development Team	Product Owner	4	MEDIUM	
9	Which review type is the M... Walkthrough	Inspection	Pair programming	Informal desk check	2	MEDIUM	
10	What is the main purpose... To describe how the GUI...	To model classes their attri...	To specify boundary value...	To show the project plan a...	2	MEDIUM	
11	A JUnit test that checks the... A unit test	A system test	An acceptance test	A usability test	1	MEDIUM	
12	In the context of stubs and ... A simple implementation of	A tool that monitors system...	A test code that calls the c...	A database used for autom...	3	MEDIUM	
13	Which test design technique... Equivalence Partitioning	Boundary Value Analysis	State Transition Testing	Random Testing	2	HARD	
14	In Agile, which statement a... Iterations happen only afte...	Iterations are long phases	Each iteration results in po...	Testing is postponed until t...	3	MEDIUM	
15	Which of the following is N... Defects are found earlier	Merge conflicts are reduced	It removes the need for uni...	Build and test processes a...	3	HARD	
16	During an Inspection, whic... Reader	Author	Moderator	Recorder	3	MEDIUM	
17	What is the main goal of ar... To optimize the GUI colors	To define how system resp...	To write detailed test cases	To estimate the project bu...	2	HARD	
18	Which metric BEST indicat... High number of bugs found	High percentage of statem...	High number of GUI screen	Low execution time of the t...	2	HARD	
19	You have a function that re... Random Testing only	Equivalence Partitioning pl...	State Transition Testing only	Ad-hoc Testing by an expe...	2	EXPERT	
20	A team maintains a large p... Commit everything directly	Use feature branches and	Use only one shared bran...	Push code to a USB backu...	2	EXPERT	
21	Which combination BEST ... Verify code style and nam...	Verify code style and nam...	Verify that each class com...	Measure low-level algorith...	2	EXPERT	
22	Which White-Box technique... Decision or Condition Cov...	Random functional tests	Usability testing	Alpha/Beta testing	1	EXPERT	
23	Which statement about T... Tests are written after full i...	TDD encourages writing te...	TDD eliminates the need f...	TDD is used only for GUI...	2	EXPERT	
24	A serious defect was found... Adding more informal chats	Performing root-cause ana...	Asking developers to work	Removing all unit tests	2	EXPERT	

[Add](#) [Edit](#) [Delete](#) [Save](#)

Difficulty: All | Correct: All | ID: Apply | Clear

[Add Question](#)

Add New Question

ID:	<input type="text"/>
Text:	<input type="text"/>
Option A:	<input type="text"/>
Option B:	<input type="text"/>
Option C:	<input type="text"/>
Option D:	<input type="text"/>
Correct:	<input type="text"/>

[Cancel](#) [Save](#)

ID	Text	A	B	C	D	Correct	Difficulty
2	In Agile development, what... A detailed technical design.	3	EASY				
3	What does the acronym S... Software Review Specifica...	2	EASY				
4	Which artifact is typically pr... Unit test cases	4	EASY				
5	Which of the following BE... Testing based on the inter...	2	EASY				
6	What is the goal of a Code... To rewrite the code from s...	3	EASY				
7	Which of the following is T... It ignores the internal struct...	3	MEDIUM				
8	In Scrum, who is mainly res... Stakeholders	4	MEDIUM				
9	Which review type is the M... Walkthrough	2	MEDIUM				
10	What is the main purpose... To describe how the GUI...	2	MEDIUM				
11	A JUnit test that checks the... A unit test	1	MEDIUM				
12	In the context of stubs and ... A simple implementation of	3	MEDIUM				
13	Which test design technique... Equivalence Partitioning	2	HARD				
14	In Agile, which statement a... Iterations happen only afte...	3	MEDIUM				
15	Which of the following is N... Defects are found earlier	3	HARD				
16	During an Inspection, whic... Reader	3	MEDIUM				
17	What is the main goal of ar... To optimize the GUI colors	2	HARD				
18	Which metric BEST indicat... High number of bugs found	2	HARD				
19	You have a function that re... Random Testing only	2	EXPERT				
20	A team maintains a large p... Commit everything directly	2	EXPERT				
21	Which combination BEST ... Verify code style and nam...	2	EXPERT				
22	Which White-Box technique... Decision or Condition Cov...	1	EXPERT				
23	Which statement about T... Tests are written after full i...	2	EXPERT				
24	A serious defect was found... Adding more informal chats	2	EXPERT				

[Add](#) [Edit](#) [Delete](#) [Save](#)

games history

Difficulty: All | Result: All | Search: [Search](#)

Players	Date / Time	Difficulty	Result	Final Score	Remaining Lives	Correct Answers	Accuracy	Duration
Maria + Or	12/12/25 21:51	EASY	LOST	69	0	0/0	-	00:37
Or + Alla	12/12/25 21:54	EASY	LOST	73	0	0/2	0%	01:38
ds + sd	16/12/25 14:27	EASY	LOST	84	0	0/3	0%	01:10
zxc + zxvc	19/12/25 20:21	EASY	WON	162	7	0/1	0%	03:25
acsdf + scdfv	19/12/25 20:28	EASY	LOST	106	0	0/0	-	01:18
Maria + Or	20/12/25 16:15	EASY	LOST	154	0	1/1	100%	06:37
Kira + Lily	21/12/25 12:45	EASY	WON	147	2	7/7	100%	04:57
dfcgvf + dxcfgv	21/12/25 14:52	EASY	LOST	136	0	2/2	100%	01:07
m. + _	21/12/25 23:09	EASY	LOST	117	0	0/0	-	01:32
sdcf + ds	01/01/26 19:02	EASY	LOST	128	0	0/0	-	01:12
df + xf	01/01/26 19:40	EASY	LOST	126	0	0/0	-	00:56
Luka Ruk	09/01/26 21:04	MEDIUM	LOST	130	0	1/0	100%	00:46

Player	Total Games	Best Score	Average Accuracy	Preferred Difficulty
Rema	1	75	0%	MEDIUM
Sasha	2	223	50%	MEDIUM
scdf	1	128	0%	EASY
df	1	126	0%	EASY
_	1	117	0%	EASY
Mili	1	211	0%	MEDIUM
Or	4	154	33%	EASY
scdf	1	106	0%	EASY
Shrel	1	148	75%	MEDIUM
dxcfgv	1	135	100%	EASY
zxvc	1	162	0%	EASY
Luka Ruk	2	130	75%	EASY

Iteration 3

7. סיום המערכת ותוכנות מתקדמות

סעיף זה מפרט את השיפורים שנוסףו באיטרציה 3, תוך התמקדות בשמירת נתונים, תוכנות ניהול ומבנה עיצוב ארכיטקטוניות.

7.1 אופן הרצה המערכת (עדין) שיטת הרצה:

לא שינוי מסעיף 2. המערכת עדין מופעלת באמצעות קובץ JAR או מחלקת MainFrame.

7.2 שימרת נתונים:

בהפעלה הראשונה, המערכת יוצרת אוטומטית תיקייה אחסון מקומית (.scorpion-minesweeper/data). בתיקייה הבית של המשתמש כדי לאחסן את הקובץ game_history.csv ומגררי שאלות מותאמים אישית. זה מבטיח שהנתונים ישמרו בין הפעולות.

7.3 קלטים למערכת (תוכנות חדשות) בנוסף לקלטי המשחק שתוארו בסעיף 3, איטרציה 3 מטפלת בקלטים המתקדים הבאים:

7.3.1 סינון היסטוריה: מונח חיפוש: קלט טקסט לסינון היסטוריה לפי שם שחזור. מסנן רמת קושי: בחירה מתוך רשימה (קל, בינוני, קשה, הכל). מסנן תוצאה: בחירה מתוך רשימה (ניצחון, הפסד, הכל).

7.3.2 ניהול שאלות מנהל: מטא-נתונים של שאלות: קלטים עבור טקסט השאלה, 4 אפשרויות, תשובה נכונה (D-A), ורמת קושי. פעולות קבצים: פקודות ל"שמירה" של שאלות שונות בחזרה לאחסן ה-CSV הקבוע.

7.4 פלטים צפויים (התנהוגיות חדשות) יצרת נתונים קבועים: פلت CSV: המערכת כעת מייצרת וمعدכנת קובץ game_history.csv. כל משחק שהסתיים מוסיף שורה חדשה המכילה: חותמת זמן, שחקנים, רמת קושי, תוצאה, ניקוד, חיים, משך ותיקון.

7.4.1 עדכוני מצב בזמן אמת (תבנית צופה): התצוגה העילית (HUD) מעדכנת כעת מידע על ניקוד ורמה באמצעות תבנית העיצוב Observer. בקר המשחק (הנושא) מודיע ללוח המשחק (הצופה) מיד עם שינוי מצב, ומבטיח אפס השהייה במשוב ממשך.

7.4.2 תוכן דינמי: מנהל השאלות מבטיח כעת שאלות ייחודיות לכל סשן משחק למניעת חזרתיות, וטוען מחדש את המאגר באופן דינמי במידת הצורך.

7.5 פעולות משתמש קצה (מתקדם) המשתמש יכול כעת לבצע את הפעולות הנוספות הבאות:

- 7.5.1 ניתוח היסטוריה: סינון משחקים קודמים כדי לראות רק ניצחונות או רמות קושי ספציפיות. צפיה בסטטיסטיות משחקן (סך הכל משחקים, ניקוד הטוב ביותר, דיק) הנגזרות מנתוני ההיסטוריה הגלומיים.
- 7.5.2 ניהול תוכן (מנהל): אימות באמצעות סיסמה (ADMIN). הוספה שאלות טריוויה חדשות למשחק. עERICA או מחיקה של שאלות קיימות לה适应ת חווית המשחק.
- 7.6 בדיקות מערכת (איטרציה 3) הבדיקות בשלב זה התמקדו בשמרנות נתונים, תבניות עיצוב וintégrציה.
- 7.6.1 בדיקות אוטומטיות (unit) בבדיקה שמירה (GameHistoryManagerTest): אימות שקובץ game_history.csv נוצר אם הוא חסר. אימות שתוצאות משחק מתואספות כראוי וניתנות לטעינה חדשה לאחר הפעלה מחדש של האפליקציה. בדיקת תווים מיוחדים (פסיקם/מרכזאות) במקרים מהבטעת תקינות ה-CSV.
- 7.6.2 בדיקות לוגיקה של מנהל (QuestionManagerTest): אימות שהוסף שאלה מעדכנת את הרשימה בזיכרון באופן מיידי. אימות ש"שמירה" כתבת שינויים לדיסק בצורה תקינה.
- 7.6.3 בדיקות סינגולטון וצופה: אימות ש-GameController.getInstance() מחזיר תמיד את אותו אובייקט. אימות שמאזינים שנרשמו דרך registerObserver מקבלים עדכונים כאשר הnickod משתנה.
- 7.6.4 בדיקות אינטגרציה מחזור משחק מלא: שיחקנו משחק מלא -> ניצחון -> אימות שהרשומה מופיעה ב"צפיה בהיסטוריה" באופן מיידי. מחזור מנהל-لמשחק: הוספה "שאלת בדיקה" חדשה כמנהל -> התחלת משחק חדש -> אימות שהשאלה החדשה מופיעה בתא שאלה.

תודה רבה ,
צוות סקורפיון

