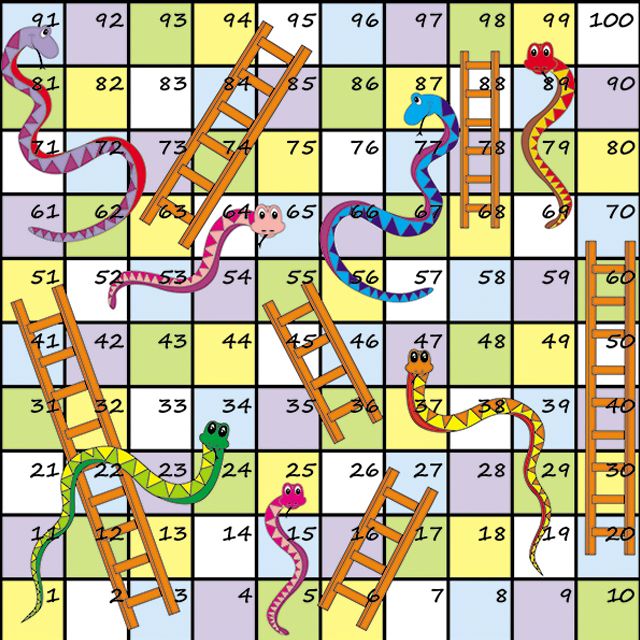
***פרויקט קבוצתי בפיתוח זריז בהנדסת תוכנה ואבטחת איכות תוכנה במערכות מידע סולמות וחבלים***

**מטלה 2: מסמך דרישות - Software Requirements Specification**

**snakes and ladders**



**מוגש על ידי: קבוצת panda**

עדן עוראבי יארא גבן

גידאא עזאם יומנה זועבי

**תוכן עניינים:**

**1.Introduction----------------------------------------------------------- 1-2**

Purpose 1.1 ------------------------------------------------------ 1 (ע"י יארא)

1.2 Scope---------------------------------------------------------1 (ע"י גידאא ויארא)

1.3 Definitions, acronyms, and abbreviations------------ 1-2 (ע"י גידאא)

1.4 References--------------------------------------------------- 2 (ע"י יומנה)

1.5 Overview------------------------------------------------------2 (ע"י עדן)

**Overall description 2---------------------------------------------------- 2-7**

2.1 -------------------------------Product Perspectiv-------- 2-5 (ע"י גידאא ויארא)

System Interfaces 2.1.1 ----------------------------------- 3

User Interface 2.1.2 ----------------------------------------4 -3

2.1.3 Hardware interfaces-------------------------------- 4

2.1.4 Software interfaces--------------------------------- 4

Communications Interfaces 2.1.5 ----------------------- 4

2.1.6 Memory Constrains---------------------------------- 4

Operations 2.1.7 --------------------------------------------- 4-5

2.1.8 Site adaptation requirements---------------------- 5

2.2 Product functions--------------------------------------------- 5-6 (ע"י עדן)

2.3 User characteristics-----------------------------------------6 (ע"י עדן)

2.4 Constraints----------------------------------------------------- 6-7 (ע"י עדן ויומנה)

2.4.1 Regulatory policies----------------------------------- 6

2.4.2 Hardware limitations--------------------------------- 6

2.4.3 Interfaces to other applications------------------- 6

2.4.4 Parallel operation------------------------------------ 6

2.4.5 Audit functions ---------------------------------------- 6

2.4.6 Control functions-------------------------------------- 6

2.4.7 Higher-order language requirements------------- 6

2.4.8 Signal handshake protocols------------------------ 7

2.4.9 Reliability requirements------------------------------ 7

2.4.10 Criticality of the application ----------------------- 7

2.4.11 Safety and security considerations-------------- 7

Assumptions and dependencies 2.5 -------------------------- 7 (ע"י יומנה)

Apportions and dependencies 2.6 --------------------------- 7 (ע"י יומנה)

**Specific requirements 3------------------------------------------ 7-15**

External interface requirements 3.1 ------------------------ 7-9

3.1.1 User interfaces ----------------------------------- 7-8 (ע"י יומנה וגידאא)

Hardware interfaces 3.1.2 ------------------------------ 9 (ע"י יארא)

Software interfaces 3.1.3 -------------------------------- 9 (ע"י עדן ויארא)

Communications interfaces 3.1.4 --------------------- 9 (ע"י עדן)

Classes/Objects 3.2 --------------------------------------- 9-14 (ע"י כל חברי הקבוצה)

3.3 Performance Requirements----------------------------- 14 (ע"י יומנה)

3.4 Design Constraints---------------------------------------- 15 (ע"י גידאא)

Software system attributes 3.5 ------------------------------15 (ע"י יארא ועדן)

Reliability 3.5.1 --------------------------------------------- 15

3.5.2 Availability ------------------------------------------ 15

3.5.3 Security --------------------------------------------- 15

Maintainability 3.5.4 -------------------------------------- 15

Portability 3.5.5 -------------------------------------------- 15

3.6 15---------------------------------------Other Requirements(ע"י גידאא)

**1.Introduction**

**Purpose 1.1**

מטרת המסמך היא לאפיין את הדרישות של המשחק "סולמות וחבלים".

המסמך מפרט את המאפיינים של המשחק ואופן השימוש בו כולל החוקים שלו, הוא מפרט את הדרישות הפונקציונאליות של המשחק בנוסף לדרישות עבור כל אובייקט שנכלל במערכת. המטרה העיקרית למערכת היא לספק למשתמשים כלי בידור.

קהל היעד של המסמך הם צוות הפיתוח של התוכנה , מנהל המוצר וצוות היישום.

**1.2 Scope**

המסמך מאפיין את המשחק הממוחשב "סולמות וחבלים", במשחק ישתתפו מ 2 עד 4 שחקנים שבוחרים באחת משלושת הרמות של המשחק. כל שחקן צריך לזוז בצורה אופקית תוך לוח המשחק לפי תוצאת הטלת הקובייה, בנוסף יש גורמים אחרים שמשביעים על תזוזת השחקן כמו סולם שגורם לשחקן לזוז באופן אנכי כלפי מעלה ,ונחש שגורם לשחקן לרדת באופן אנכי .בנוסף קיים סוגים שונים של משבצות הפתעה/זכייה.. שמשפיעות על הפעולה הבאה של השחקן .תוך כדי המשחק השחקן יצטרך לתת מענה נכון על שאלות שקשורות לתחום הנדסת תוכנה הוא יקבל שאלות אלה במשבצת שאלה או בתוצאת הקובייה, לכן המשתתפים צריכים להיות בעלי ידע בנושא כדי שיענו נכון על השאלות ויוכלו להתקדם במשחק, המטרה של כל שחקן היא להגיע ראשון למשבצת האחרונה בלוח.

המשחק מהווה כלי בידור לשחקנים ומעודד תחרות ביניהם ובאותו זמן עוזר להם לרענן את הידע שלהם בתחום הנדסת תוכנה.

**1.3 Definitions, acronyms, and abbreviations**

* Java: שפת תכנות ברמה גבוהה, מבוססת על מחלקות ומונחת עצמים.
* GUI-Graphical User Interface: ממשק משתמש לתוכנה, המבוסס על עיצוב גרפי של המסך המוצג למשתמש.
* Eclipse: סביבת עבודה לפיתוח יישומים והיא תומכת בשפת התכנות Java.
* Swing: ממשק משתמש גרפי של ערכת כלים ל JAVA
* GitHub: שירות ניהול גרסאות ושירות אחסון, מבוסס רשת, עבור מיזמי פיתוח תוכנה.
* JSON: תקן של פורמט קובץ ופורמט העברת מידע שמשתמש בטקסט קריא-לאדם כדי לאחסן ולהעביר מבני מידע המורכבים מזוגות של מפתח-ערך.
* MVC -Model View Controller: היא תבנית עיצוב בהנדסת תוכנה המשמשת להפשטת יישום כלשהו. התבנית מתארת טכניקה לחלוקת היישום לשלושה חלקים: "מודל", "תצוגה" ו"בקר", המחוברים ביניהם בצימוד רפוי מונחה אירועים.

**1.4 References**

**מפרט דרישות**-**תקן:** 830 IEEE (IEEE Std 830-1998)

IEEE Recommended Practice for Software Requirements Specifications""

**תאריך**: Approved 25 June 1998

**ארגון הוצאה לאור**: Software Engineering Standards Committee of the IEEE Computer Society

**מקור- אתר הקורס-קישור:** <https://mw24.haifa.ac.il/pluginfile.php/194224/mod_resource/content/1/IEEE830.pdf\>

**1.5** **Overview**

**החלק השני** של מסמך הדרישות מתייחס באופן כללי למערכת ודרישותיה, לפונקציונליות שלה, למי היא מיועדת, אילוצי המערכת ואפיון ממשק משתמש. **החלק השלישי** מסביר באופן מפורט את דרישות המערכת ומתייחס באופן ספציפי לפונקציונליות ולהתנהגות המערכת בנוסף לממשק המשתמש, חומרת המערכת. וזה תוך שימוש במונחי מפתחי מערכת.

**Overall description 2**

**2.1 Product Perspective**

המשחק "סולמות וחבלים" הוא משחק מרובה משתתפים משחקים בו מ 2 עד 4 שחקנים,

השחקנים יופיעו בצורת אייקון ולכל שחקן יהיה צבע ייחודי. המשחק מתנהל על גבי לוח משחק שמחולק למשבצות ממוספרות ומשבצות מיוחדות והוא מרושת עם סולמות ונחשים המחברים משפצות ספציפיות.

המטרה של כל שחקן היא לנווט החל ממשבצת 1 ועד משבצת N שערכו תלוי ברמת המשחק, על ידי הטלת קובייה שקובעת את הצעדים להתקדמות או מגרילה לו שאלה בנוסף לטיפוס על סולמות והימנעות מנחשים שמורידים את השחקן.

גודל הלוח מוגדר לפי רמת המשחק במידה ונבחרה רמה קלה הלוח יהיה בגודל 49 משבצות , רמה בינונית :100 משבצות ,רמה קשה:169 משבצות .

מטרת הקובייה היא לקבוע פעולת השחקן בתור הנוכחי ,בקובייה אפשר לקבל שאלה או מספר צעדים להתקדמות .סיכויי קבלת שאלה בקובייה ומספר הצעדים להתקדמות נקבעים לפי רמת המשחק .

בנוסף למשבצות ממספרות ישנם משבצות מיוחדות כמו : משבצת שאלה שמגרילה שאלה עבור השחקן , משבצת הפתעה קפיצה 10 צעדים אחורה/ קדימה. מספר המשבצות מכל סוג נקבע לפי רמת המשחק .

השאלות במשחק הם שאלות אמריקאיות בנושאי הקורס ולכל שאלה ישנם 4 תשובות אפשריות ו- 3 רמות (קלה/בינונית/קשה) .מענה נכון או שגוי על שאלה גורם לתזוזת השחקן לפי רמת השאלה.

במשחק ישנם סוגים שונים של נחשים ולכל סוג יש צבע ואורך שונה , כמות הנחשים מכל סוג נקבעת לפי רמת המשחק .

לסולמות שבמשחק יש אורכים שונים , כמות הסולמות מכל סוג נקבעת לפי רמת המשחק .

מיקום האובייקטים שמפוזרים על גבי לוח המשחק כמו הסולמות , הנחשים, משבצות מיוחדות ייקבע באופן רנדומלי בתחילת כל משחק.

למשחק יש מאגר שאלות שמנוהל באמצעות המערכת כך שאפשר לערוך/להוסיף/למחוק שאלות , בנוסף המשחק ישמור היסטוריה ויתמוך בצפייה בהיסטוריית משחקים קודמים.

**System Interfaces 2.1.1**

המערכת מנהלת משחק "הסולם והחבלים" שמותקן באופן מקומי על מחשב המשתמש. הממשק מנהל את זרימת המשחק לפי החוקיות שלו והמשתמש יוכל לצפות בלוח המשחק ו יוכל לראות את כל האלמנטים שמעורבים במשחק... ויקבל משוב והודעות שיופיעו על המסך. המערכת תשלוף את השאלות מתוך קובץ Json ותשמור שאלות חדשות בקובץ זה ותשתמש בקובץ טקסט כדי לשמור את היסטוריית המשחקים.

**User Interface 2.1.2**

הממשק מכיל מסך ראשי שמכיל 4 כפתורים ועל כל כפתור יש Label שמסביר לאן הכפתור מנווט את המשתמש. הכפתורים הם 1) "התחלת משחק" 2)" חוקי המשחק" 3) "צפייה בהיסטוריה" 4) "ניהול שאלות". בעת הלחיצה על "התחלת משחק" המשתמש ייבחר במספר השחקנים ורמת המשחק . המסך הבא הוא הזנת שמות השחקנים ובחירת צבע.

במידה והמשתמש מנסה לעבור לשלב מסוים לפני הזנת הערכים של השלב הקודם אז תופיע לו הודעת שגיאה ויצטרך למלא את הערכים.

במסך הבא - מסך המשחק יוצג לוח משחק ממוספר ומחולק למשבצות בגודל שמתאים לרמת המשחק ויוצגו כל הישויות המעורבות במשחק, גם כן יוצג לנו איפה כל שחקן עומד ואת התנועה שלו במהלך המשחק (עליה ,ירידה והליכה) . במידה והשחקן מקבל שאלה אז יקפוץ לו חלון על המסך שמכיל שאלה אמריקאית עם 4 תשובות ואחרי שהשחקן ייבחר בתשובה יקבל משוב מתאים .

במהלך המשחק יוצג מעל הלוח הודעות עדכניות כמו הודעות על התור הנוכחי, תוצאת הקובייה ,כלומר יהיה תמיד משוב לשחקנים על כל פעולה שהם מבצעים...

בכל מסך יש לנו כפתור חזרה למסך הקודם בנוסף לכפתור חזרה לעמוד הראשי .

הממשק נבנה בסביבת העבודה Eclipse עם שימוש בספריית Swing, הממשק נוח לשימוש וידידותי למשתמש כי הוא תמיד נותן מידע ומשוב ויזואלי שעוזר למשתמש לעקוב אחרי הפעילויות שעשה בנוסף הוא מונע טעויות על ידי הגבלת המשתמש.

**2.1.3 Hardware interfaces**

המערכת תותקן מקומית על מחשב סטנדרטי שתומך בשפת התכנות 8Java, ותוצג על המסך ותופעל באמצעות מקלדת ועכבר.

**2.1.4 Software interfaces**

המערכת תכתב בשפת 8Java ותפותח בסביבת העבודה Eclipse

יחד עם שימוש בספריית Swing לצורך בניית ה GUI. שמירת הנתונים של מאגר השאלות תהיה בקבצי Json והיסטוריית המשחק תישמר בקבצי טקסט.

**Communications Interfaces 2.1.5**

המערכת אינה מקוונת ורק בפעם הראשונה שמורידים אותה נצטרך להתחבר לאינטרנט , וכל עוד התקנו אותה על מחשב נוכל להשתמש בה באופן מיקומי בלי צורך לאינטרנט.

**2.1.6 Memory Constrains**

* מאגר השאלות של המשחק יישמר בקובץ Json, והיסטוריית המשחקים תשמר בקבצי טקסט. הפעלת המשחק מצריכה שימוש בזיכרון המקומי של המחשב RAM.

**Operations 2.1.7**

* צפייה במסך ראשי : 3 גבוהה
* התחלת משחק
* בחירת מספר השחקנים
* בחירת רמת המשחק
* הזנת שם כינוי
* בחירת אובייקט\צבע
* צפייה בהיסטוריית משחקים
* צפייה באשף שאלות
* הוספת שאלה
* מחיקת שאלה
* עדכון שאלה (תשובה/שאלה)
* חזרה למסך הראשי
* חזרה למסך הקודם
* צפייה בלוח משחק
* הזזת השחקן קדימה
* הזזת השחקן אחורה
* עליה אנכית למעלה בעת עמידה על תחתית סולם (עליית השחקן בסולם)
* ירידה אנכית למטה בעת עמידה על ראש נחש (ירידת השחקן בנחש)
* הטלת קובייה
* צפייה בתוצאת הקובייה
* בחירת תשובה לשאלה
* סיום משחק
* צפייה בחוקי המשחק
* מעבר למסך הבא
* חישוב כמות הצעדים של השחקן
* צפייה במשוב (משוב לגבי התור הנוכחי/השם של המנצח)
* עמידה על משבצת מיוחדת והתנהגות בהתאם.

**2.1.8** **Site adaptation requirements**

המערכת שלנו מותקנת מקומית על המחשב ואין צורך להתאים את הדרישות לאתר.

**2.2 Product functions**

* בניית משחק לפי רמת הקושי ומספר השחקנים.
* בניית קובייה ,ניהול אפשריות הקובייה (חישוב לפי סטטיסטיקות).
* ניהול תור השחקנים .
* מעקב ועדכון התנהגות ותזוזת ישות השחקן.
* הצגת מסרים מתאימים.
* שליפה, שמירה, עדכון, הוספה ומחיקת שאלות.
* שמירה, שליפה לנתוני היסטוריית המשחקים.

**2.3 User characteristics**

המשחק המקורי מיועד לכל האנשים שיש להם ידע בסיסי בשימוש במחשב . אבל, המשחק שנייצר יהיה מיועד יותר לאנשים בעלי ידע בהנדסת תוכנה בעיקר לסטודנטים למערכות מידע בקורס הנדסת תוכנה ואבטחת איכות משום שהשאלות מבוססות על ידע בקורס.

**2.4 Constraints**

**2.4.1** **Regulatory policies**

עמידה בחוקי ונהלי הקורס ועמידה בחוקי המשחק ודרישותיו.

**2.4.2** **Hardware limitations**

המחשב המקומי שעליו מותקנת המערכת צריך לתמוך ב JAVA 8 ולכלול מסך, עכבר ומקלדת כדי לאפשר אינטראקציה עם המשתמש.

בנוסף לזיכרון לאחסון ועיבוד נתוני המערכת.

**2.4.3 Interfaces to other applications**

לא רלוונטי, כי למערכת שלנו אין אינטראקציה עם תוכנות אחרות.

**2.4.4 Parallel operation**

לא רלוונטי, כי במערכת שלנו אין פעולות שמתבצעות באותו זמן.

**2.4.5** **Audit functions**

לא רלוונטי, המערכת אינה דורשת הקלטת אודיו מהמשתמש.

**2.4.6** **Control functions**

במערכת נדרש בקרה על הנתונים שהמשתמש מזין אותם כמו: בקרה על כינוי ייחודי, מניעת הוספת שאלה ריקה ובדיקת הוספת 4 תשובות אפשריות לשאלה.

**2.4.7** **Higher-order language requirements**

השתמשנו בשפת התכנות המונחת עצמים Java.

**2.4.8** **Signal handshake protocols**

התוכנה עובדת על מחשב מקומי שתומך ב JAVA לכן היא לא מצריכה חיבור ופרוטוקולי אינטרנט.

**2.4.9** **Reliability requirements**

המשחק אמין מכיוון שהוא מבצע פעולות בצורה אמינה ,ההגדרות של המשחק זהות לכל השחקנים (הצעדים מחושבים באופן נכון) ויש גיבוי לנתונים ושמירת מידע.

**2.4.10 Criticality of the application**

המשחק אינו קריטי ברמה גבוהה, כי המשחק אינו כולל מידע רגיש. בנוסף לכך, המשחק שלנו מיועד בעיקר למסגרת הקורס ולא לקהל רחב.

**2.4.11 Safety and security considerations**

אין צורך לאבטחת או להצפנת מידע מכיוון שהמערכת לא כוללת מידע רגיש.

* 1. **Assumptions and dependencies**

המערכת תפעול על כל מערכת הפעלה שתומכת ב Java8 ,קבלת הקלט תהיה באמצעות מקלדת ועכבר.זמינות של יכולות עיבוד גרפי בסיסיות במכשיר המשתמש.

**Apportions and dependencies 2.6**

למערכת הזאת אין גרסאות עתידיות לכן לא יהיה דרישות עבור גרסאות אחרות.

**Specific requirements 3.**

**External interface requirements 3.1**

**3.1.1 User interfaces**

1. מסך ראשי :במסך הראשי מופיע תפריט שמורכב מ-4 כפתורים שמאפשרים למשתמש להתחיל משחק חדש, לצפות בחוקי המשחק, לנהל את השאלות , לצפות בהיסטוריית משחקים ,כל כפתור מכיל כותרת שמאתרת לאן הכפתור מנווט.

**דירוג :** 3 גבוהה ,כי פעולות אלה הן חלק בלתי נפרד ועיקרי במערכת והן נקודת ההתחלה שמנווטת את המשתמש לשאר המסכים.

1. מסך צפייה בחוקי המשחק : המסך הזה מפרט לשחקנים את חוקי המשחק.

**דירוג :** 2 בינוני , המשחק הוא משחק נפוץ וסטנדרטי ולכן יש פחות אנשים שלא מכירים אותו .

1. מסך הגדרת משחק: בעת לחיצה על כפתור תחילת משחק המשתמש יועבר למסך שבו צריך לקבוע את מספר השחקנים דרך תיבת בחירה, התיבה קובעת את מספר השחקנים כ-2 כברירת מחדל במידה והמשתמש לא בחר ערך , בנוסף לכך המסך מציג למשתמש את רמות המשחק דרך 3 כפתורי בחירה .

המסך מכיל כפתור שמעביר את המשתמש למסך הבא במידה ובחר את רמת המשחק , במידה ולא אז הוא מציג לו הודעת שגיאה שחסר לו פרטים ולא מעביר אותו הלאה, המסך מכיל כפתור חזרה למסך הראשי.

**דירוג :** 3 גבוהה, כי פעולות אלה עיקריות להגדרת פרטי המשחק.

1. מסך הזנת פרטי השחקנים : המסך מציג מספר תיבות טקסט לבחירת שם כינוי ותיבות בחירת icon לכל השחקנים, וכפתור מעבר לתחילת המשחק. המסך מכיל כפתור חזרה למסך הראשי.

**דירוג :** 3 גבוהה ,כי פעולות אלה עיקריות לקביעת נתוני השחקנים והתחלת המשחק.

1. מסך המשחק : המסך הזה מכיל את לוח המשחק וכל האובייקטים שמעליו והקובייה, ה-icons של השחקנים והתנועה שלהם בכל שלב, עם הופעת משפט מעל הלוח שמודיע לאיזה שחקן שייך התור .

במידה והשחקן מקבל שאלה לפי חוקי המשחק אז קופץ חלון עם שאלה ו 4 תשובות התשובה של השחקן על השאלה משפיעה על ההתקדמות שלו.

בכל זמן נתון השחקנים יכולים לעצור את המשחק על ידי כפתור סיום המשחק שמחזיר אותם למסך הראשי.

**דירוג :** 3 גבוהה, כי פעולות אלה הם עיקריות כך שמאפשרות למשתמש לשחק ולצפות במשחק ומאפשרת לו לענות על שאלות בנוסף מאפשרת לו לעצור את המשחק .

1. מסך צפייה בהיסטוריה : המסך הזה מאפשר למשתמש לצפות בנתוני היסטוריית המשחקים שמופיעים בטבלה, המסך מכיל כפתור חזרה למסך הראשי.

**דירוג :** 1 נמוך, כי פעולה זו היא לא קריטית ולא משפיעה על התנהלות המשחק.

1. מסך ניהול שאלות : המסך הזה מאפשר לצפות בשאלות דרך טבלה , ונותן האפשרות למשתמש לעדכן , למחוק ולהוסיף רשומות לטבלה דרך בחירה בכפתור המתאים , המסך מכיל כפתור חזרה למסך הראשי.

**דירוג :** 3 גבוהה, כי פעולה זו קריטית לניהול השאלות שמשפיעות על התנהגות השחקן.

**Hardware interfaces 3.1.2**

* החומרה שהתוכנה צריכה היא מחשב עם מסך , מקלדת ועכבר .

המסך משמש לצפייה במסכי המשחק/ממשק והעכבר משמש לביצוע אינטראקציה עם המשחק וללחוץ על הכפתורים השונים. גם המקלדת תשמש כאמצעי קלט.

**Software interfaces 3.1.3**

* התוכנה תרוץ על כל מערכת הפעלה שתומכת ב 8JAVA.
* הפרויקט ייכתב בשפת התכנות JAVA , בסביבת העבודה Eclipse.
* ממשק המשתמש יפותח באמצעות Swing.
* GitHub ישמש לניהול ניהול התצורה, התיעוד ואחסון קבצי המשחק.
* היסטוריית המשחק/תוכנה תשמר בקבצי טקסט .
* פרטי השאלות והתשובות יישמרו בקובץ JSON עם סכמה קבועה.
* חלוקת הארכיטקטורה של התוכנה תהיה לפי MVC.

**Communications interfaces 3.1.4**

התוכנה עובדת על מחשב מקומי שתומך ב JAVA ולא מצריכה חיבור לאינטרנט.

**Classes/Objects 3.2 :** בחרנו בתבנית זו כי המערכת שלנו מכילה אובייקטים שונים, וחלוקה זו עוזרת לנו להבין את התמונה השלמה ואת הפונקציונאליות של המערכת ומקשרת בין כל האלמנטים כך שלכל אלמנט יש מחלקה שמכילה תכונות ומתודות ומסרים.

**SysData 3.2.1 :** מחלקה לניהול נתוני ההיסטוריה ומאגר השאלות**.**

**Attributes 3.2.1.1:**

* Games : מבנה נתונים ששומר אובייקטים מסוג .Games
* Questions: מבנה נתונים ששומר אובייקטים .Questions

**Functions 3.2.1.2:**

* UploadGames: מתודה ששולפת את פרטי המשחקים הקודמים מקובץ טקסט ושומרת אותם במבני נתונים Games.
* UploadQuestions: מתודה ששולפת את פרטי השאלות מקובץ הJson ושומרת אותם במבנה הנתונים Questions.

**:Messages 3.2.1.3** אין מסרים .

**Game 3.2.2 :** מחלקה ששומרת את פרטי המשחק בנוסף לפרטי לוח המשחק.

**Attributes 3.2.2.1**

* GameId :מספר ייחודי שמזהה את המשחק .
* GameLevel: משתנה ששומר את רמת המשחק הנוכחי.
* StartTime :משתנה מסוג Date/Time ששומר את זמן התחלת המשחק.
* EndTime :משתנה מסוג Date/Time ששומר את זמן סיום המשחק.
* RowsNum :מספר השורות של הלוח.
* ColsNum :מספר המשבצות בכל שורה בלוח.
* Board :מבני נתונים שבגודל RowsNum\*ColsNum ששומר את לוח המשחק.
* WinnerId: מספר ייחודי שמזהה את השחקן המנצח במשחק.
* CurrentTurn: מספר מזהה ייחודי של השחקן שהוא בעל התור הנוכחי .
* Players : מבני נתונים ששומר אובייקטים מסוג Playerשל השחקנים.
* Dice: אובייקט מסוג Dice ששומר פרטי קוביית המשחק.
* Snacks: מבני נתונים ששומר אובייקטים מסוג Snack .
* Ladders: מבני נתונים ששומר אובייקטים מסוג Ladder.
* Square : מבני נתונים ששומר אובייקטים מסוג Square.

**Functions 3.2.2.2**

* CreateGame: מתודהשמאתחלת את המשחק ומייצרת אובייקט מסוג Game בנוסף לזאת מייצרת אובייקטים מסוג Snackו Ladderומייצרת את המשבצות והקובייה ושומרת הסוג שלהם במבני הנתונים Board.
* StartGame : מתודה שמתחילה את המשחק וקובעת מי השחקן המתחיל.
* EndGame : מתודה שמוחקת את פרטי המשחק ממבני הנתונים.
* ShowWinner : שולפת את הנתונים של השחקן המנצח ממבני הנתונים.
* GameTime: מתודה שמחשבת את זמן המשחק .
* SaveHistory: מתודה ששומרת את פרטי המשחק במסד הנתונים.
* :GameQueue מתודה שמנהלת את התור של המשחק , היא משתמשת במבני הנתונים Players ומנהלת את התור .
* UpdatePlayerPlace**:** מתודה שמזיזה את השחקן בהתאם לתוצאה שקיבל אחרי הטלת קובייה בנוסף לזאת אם הוא עומד על משבצת שאלה\הפתעה ואם נתקל בסולם או בנחש אז היא תעדכן את קואורדינטות בהתאם.

**Messages 3.2.2.3**

* המשחק התחיל !
* התור הוא של שחקן \_\_\_
* המנצח הוא \_\_\_
* המשחק הסתיים !
* תוצאת הקובייה \_\_\_

**3.2.3 Square :** מחלקה ששומרת מידע על המשבצת.

**Attributes 3.2.3.1:**

* SquareID: מספר ייחודי שמזהה מספר כל משבצת.
* SquareRow: מספר השורה של המשבצת.
* SquareCol: מספר העמודה של המשבצת.

**Functions 3.2.3.2**: אין מתודות**.**

**Messages 3.2.3.3:** אין מסרים**.**

**4.2.3 QuestionSquare :**

מחלקה זו היא מחלקה יורשת ממחלקת Square, והיא שומרת מידע על משבצת השאלה.

**Attributes 3.2.4.1:**  אין תכונות.

**Functions 3.2.4.2:**

* ChooseQuestion: מתודהשמגרילה את רמת השאלה אחר כך מגרילה שאלה מתוך מאגר השאלות המתאים (מבני הנתונים ששומר את השאלות).

**Messages 3.2.4.3:**

* השאלה שקיבלת היא :\_\_\_\_\_\_
* התשובות האפשריות הן : \_\_\_\_\_
* התשובה שלך נכונה 😊
* התשובה שלך לא נכונה ☹

**5.2.3 SurpriseSquare :** מחלקה זו היא מחלקה יורשת ממחלקת Square, ומייצרת אובייקט מסוג משבצת ההפתעה .

**Attributes 3.2.5.1:** אין תכונות.

**Functions 3.2.5.2:**

* RandSurprise: מתודה שמגרילה הפתעה בין אם היא 10 צעדים אחורה\קדימה .

**Messages 3.2.5.3:** אתה צריך לזוז 10 צעדים (קדימה \אחורה).

**3.2.6 :Question** מחלקה לייצוג שאלה במשחק סולמות וחבלים.

**Attributes 3.2.6.1:**

* Id : מס ייחודי לכל שאלה.
* Type : רמת הקושי של השאלה .
* Content : תוכן השאלה.
* Answers : מבני נתונים שמכיל ארבע התשובות .
* TrueAnswer : התוכן של התשובה הנכונה.

**Functions 3.2.6.2:**

* addQuestion : מוסיפה שאלה חדשה לקובץ ה- Json שיש בו את כל נתוני השאלות.
* deleteQuestion : מוחקת שאלה מקובץ ה- Json.
* editAnswer : עריכת אחת התשובות השייכות לשאלה מסוימת.
* isCorrectAnswer : בודקת אם התשובה שהשחקן בחר נכונה.
* CheckSteps :בהינתן את השאלה והתשובה של המשתמש הפונקציה מחזירה לי מספר הצעדים של ההתקדמות\החזרה .

**Messages 3.2.6.3:**

* השאלה התווספה בהצלחה .
* השאלה נמחקה בהצלחה .
* התשובה עבור השאלה הזאת נערכה בהצלחה .
* התשובה שלך נכונה.
* התשובה של שגויה .
* מספר הצעדים קדימה הן : \_\_\_\_\_\_
* מספר הצעדים אחורה הן : \_\_\_\_\_\_

**3.2.7 Snake** **:** מחלקה ששומרת פרטי נחש.

**Attributes 3.2.7.1:**

* Id : מס ייחודי לכל נחש.
* Type : הצבע של הנחש.
* XTailNum : מס המשבצת שזנב הנחש מונח עליו בציר אנכי .
* YTailNum : מס המשבצת שזנב הנחש מונח עליו בציר אופקי.
* XHeadNum : מס המשבצת שהראש שלה מונח עליו בציר אנכי.
* YHeadNum – מס המשבצת שהראש שלה מונח עליו בציר אופקי.

**Functions 3.2.7.2:**

* MovePlayerDown: מתודה שמעדכנת את קואורדינטות השחקן לפי סוג הנחש .
* SnackRandom: מתודה שמגרילה מיקום הנחשים לכל סוגי הנחשים.

**Messages 3.2.7.3:** עכשיו אתה צריך לרדת בנחש ☹

**3.2.8.1 : Ladder** מחלקה ששומרת את פרטי הסולם.

**Attributes 3.2.8.1:**

* LadderId : מס ייחודי לכל סולם.
* Length : האורך של הסולם.
* Xstart : מס המשבצת שתחתית הסולם מונחה עליו בציר אנכי .
* Ystart : מס המשבצת שתחתית הסולם מונחה עליו בציר אופקי.
* Xend : מס המשבצת שראש הסולם מונח עליה בציר אנכי.
* Yend :מס המשבצת שראש הסולם מונח עליה בציר אופקי.

**Functions 3.2.8.2:**

* MovePlayerUp: מתודה שמעדכנת את קואורדינטות השחקן לפי אורך הסולם .
* LadderRandom: מתודה שמגרילה מיקום הסולמות לכל סוגי הסולמות.

**Messages 3.2.8.3:** עכשיו אתה עולה בסולם 😊

**Dice 3.2.9:** מחלקה ששומרת את פרטי הקובייה.

**Attributes 3.2.9.1:** NumOfSides : מספר הצדדים של הקובייה.

**Functions 3.2.9.2:**

* RollDice :המתודה מגרילה מספר בין 1 לבין מספר צדדי הקובייה , לפי רמת המשחק וחישוב הסטטיסטיקות.
* checkResultEasy : המתודה מופעלת כשרמת המשחק היא קלה: היא מקבלת את תוצאת הקובייה ומחזירה הפעולה הבאה של השחקן.
* checkResultMedium: המתודה מופעלת כשרמת המשחק היא בינונית: היא מקבלת את תוצאת הקובייה ומחזירה הפעולה הבאה של השחקן.
* checkResultDificult: המתודה מופעלת כשרמת המשחק היא קשה : היא מקבלת את תוצאת הקובייה ומחזירה הפעולה הבאה של השחקן.
* CallQuestion: מתודה שמגרילה שאלה שמתאימה לרמת השאלה שקיבלנו בקובייה.

**Messages 3.2.9.3:** אין מסרים.

**3.2.10 : Player** מחלקה ששומרת את פרטי השחקן.

**Attributes 3.2.10.1:**

* PlayerID: מספר ייחודי שמזהה את השחקן.
* Color: צבע שמזהה השחקן.
* Icon: אייקון שמזהה השחקן.
* PlayerCol: מספר העמודה הנוכחית שהשחקן עומד עליה.
* PlayerRow: מספר השורה הנוכחית שהשחקן עומד עליה.

**Functions 3.2.10.2:**

* UpdateRow: מתודה שמעדכנת מספר השורה הנוכחית שהשחקן עומד עליה .
* UpdateCol: מתודה שמעדכנת מספר העמודה הנוכחית שהשחקן עומד עליה .

**Messages 3.2.10.3:** אין מסרים.

**3.3 Performance Requirements**

* תגובת המערכת צריכה להיות מהירה: לחיצה על כפתורים תתבצע באופן יעיל כך שלא ייקח למשתמש יותר משניה לביצוע הפעולה.
* פעולות חישוב מיקום השחקן וההזזה שלו על הלוח צריכות להתבצע במהירות.
* צריך לשלוח למשתמש פידבק ברור ומהיר שמתאים לסוג המשבצת ו תוצאת הקובייה.
* עדכון אוטומטי להיסטוריית המשחקים ברגע סיום המשחק.
* הצגה מידית של השחקן ברגע החלפת התור.

**3.4 Design Constraints**

הממשק שלנו צריך להיות נוח , ידידותי ומספק חווית משתמש טובה דרך שימוש בכפתורים בולטים לעיין ומסודרים לפי החשיבות שלהם ומכילים טקסט שמעיד לאן הכפתור מנווט , דרך שימוש ברכיבים שמגבילים את המשתמש ולא מאפשרים הכנסת ערכים לא רלוונטיים , גם דרך שימוש בצבע ייחודי לכל שחקן שזה עוזר להם לעקוב אחרי המשחק .

**Software system attributes 3.5**

**Reliability 3.5.1**

המערכת צריכה להיות אמינה כך שהיא מבצעת פעולות בצורה אמינה (מבצעת הגרלה בצורה נכונה \מחשבת הצעדים של כל שחקן לפי הגדרות המשחק וכו..) .ומבצעת גיבוי לנתונים ושומרת מידע ומאפשרת שמירת היסטוריית משחקים .

**3.5.2**  **Availability**

המערכת צריכה להיות זמינה עבור המשתמש כך שיוכל להשתמש בה בכל עת בלי תקלות.

**3.5.3 Security**

המידע והנתונים שהמערכת שומרת כמו היסטוריית המשחקים הוא מידע לא רגיש ואין צורך לשמור על האבטחה והסודיות שלו.

**Maintainability 3.5.4**

המערכת היא תוכנה עצמאית עם פונקציות קבועות שמותקנת באופן מקומי על מחשב המשתמש לכן היא לא צריכה תחזוקה . צוות הפיתוח יעקוב תמיד אחרי פעילות המערכת ויתמוך בה במידת הצורך .

**Portability 3.5.5**

המערכת תותקן באופן מקומי על מחשב סטנדרטי והיא כתובה בשפת Java

בגרסה 8 לכן היא תפעל על כל מערכת הפעלה שתומכת ב Javaועל מחשבו האישי של המשתמש אז לא יהיה צורך לניידות.

**3.6 Other Requirements**

בשלב זה אין למערכת דרישות נוספות , במידה ויהיה לנו דרישות נוספות נעדכן אותם בהתאם.