**תרגיל 1 - קורס ג'אווה ואינטרנט**

**מגישים:**

**אמיר שביט -**

**יורי ריטבין - 316961101**

**אנחנו כאן לכל שאלה, או בקשה**

[**yuri1992@gmail.com**](mailto:yuri1992@gmail.com) **- 0508915495**

[**ashavit3@gmail.com**](mailto:ashavit3@gmail.com) **-**

**הרצת המשחק**

1. **טעינת הגדרות המשחק -** יש ללחוץ על אפשרות מספר 1, לאחר הלחיצה תתבקש לספק את שם הנתיב של קובץ הXML אותו תרצה לטעון, אם במהלך טעינת הקובץ, נתקל בבעיות אימות הנתונים, כמו למשל מספר הצוללות אינו תואם את המספר המבוקש או כל בעיית אימון אחרת, תוקפץ הודעת שגיאה, וטעינת המשחק תפסק.
2. **התחלת המשחק -** יש ללחוץ על אפשרות מספר 2, לאחר הלחיצה המשחק יחל, והשחקן הראשון ברשימה יחל את תורו הראשון.

**מהלך המשחק**

1. **הצגת סטטוס המשחק -** לחיצה על אפשרות מספר 3, תדפיס את סטטוס ממשחק הנוכחי, כלומר יודפס לוח הצללות של השחקן הנוכחי, כולל לוח התקיפות שהוא ביצע. בלוח ישנם מספר סימנים מוסכמים בלוח הצוללות של כל שחקן.
   1. **סימונים מוסכמים בלוח הצוללות:**
      1. ~ - סימון מוסכם לשטח ימי רגיל (כלומר ללא ספינה או תקיפה).
      2. @ - סימון מוסכם לצוללת הממוקמת על הלוח.
      3. % - סימון מוסכם לספינה אשר נפגעה.
   2. **סימונים מוסכמים בלוח התקיפות:**
      1. ~ - סימון לשטח ימי רגיל (כלומר ללא ספינה או תקיפה)
      2. \* - סימון לפגיעה מוצלחת בספינה
      3. ^ - סימון לתקיפה ללא פגיעה. (החטאה).
2. **ביצוע מהלך -** לחיצה על אפשרות 4, תוביל לביצוע מהלך במשחק על ידי השחקן הנוכחי, לאחר הלחיצה, על השחקן יהיה לבחור את הקורדינטה במשחק, על ידי בחירת ערך הX והY שלה בזה אחר זה. במידה והתקיימה פגיעה, השחקן התוקף ימשיך את המשחק, במידה והייתה החטאה, התור יעבור לשחקן השני.
3. **הצגת סטטיסטיקות -** לחיצה על אפשרות מספר 5, תוביל להדפסת סטטיסטיקות על המשחק הנוכחי כפי שפורטה בדרישות התרגיל.
4. **פרישה מהמשחק -** לחיצה על אפשרות מספר 6, תגרום לשחקן הנוכחי לפרוש מהמשחק, והניצחון יהיה של השחקן השני. לאחר הפרישה, המשחק יתאפס, כלומר יהיה ניתן לטעון קובץ xml חדש ולהתחיל משחק חדש.
5. **סיום המשחק -** לחיצה על אפשרות מספר 7 , תוביל לסיום מוחלט של המשחק. והתוכנית תסיים לרוץ.

**בונוסים שמומשו**

1. **לאפשר את המשך הרצת התוכנית לאחר פקודה מס' 6.**

**ארכיטקטורת המשחק**

Packages

1. Descriptor - Have the auto generated class to support JAXB methods. By design descriptor should reflect the xsd we expecting to get when loading the game.
2. Game
   1. Engine - Should contain all the core logic of battle ships game
   2. Player - Should contain implementation to all needed players we want to support on the game. (Human or Machine player)
   3. Runners - Contain the VIEW layer, so called runner, for example an Console runner will implement the game view layer using the console output.
   4. Ships - Should contain different type of ships we do want to support in the game.

Class

1. game.engine.GameManager - responsible to represent the game, manage turn, decied when do we have a winner, and be the connecting layer between players and all other parts.
2. game.engine.GameMode - enum represent for games types
3. game.engine.GameTurn - each turn played in the game is represent by this class, should know the basic information about the turn hit/miss and how much time did it take.
4. game.engine.GameStatistics - representing the collected statistics during the game.
5. game.players.Player - Representing the player of the game, each player have a ShipsBoard and AttackBoard objects to represent the Attacks he made and Hits.
6. game.players.AttackBoard - implement the attack player was made.
7. game.players.ShipsBoard - implement the located ships on the board, of each player
8. game.ships.Ship - representing a ship in the game, each ship implement his method of the positions it located at.
9. game.runners.ConsoleRunner - View implement of the game using console as output device.