**תרגיל 2 - קורס ג'אווה ואינטרנט**

**מגישים:**

**אמיר שביט -**

**יורי ריטבין -**

**אנחנו כאן לכל שאלה, או בקשה**

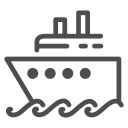
[**yuri1992@gmail.com**](mailto:yuri1992@gmail.com) **-**

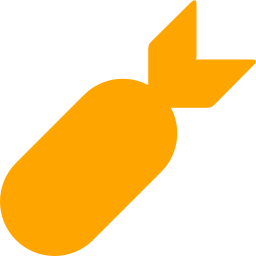
[**ashavit3@gmail.com**](mailto:ashavit3@gmail.com) **-**

**הרצת המשחק**

1. **טעינת הגדרות המשחק -** בתפריט העליון, יש ללחוץ על האפשרות - File -> Load XML, האפשרות תפתח את חלון בחירת הקבצים, ותאפשר למשתמש לבחור את קובץ הגדרות המשחק אותו הוא ירצה לטעון.
2. **התחלת המשחק -** לשם תחילת המשחק יש לבחור בתפריט העליון באפשרות File -> Start Game, לאחר לחיצה על אפשרות זאת, נראה כי המשחק מתחיל להיטען, ויוצגו לשחקן לוחות המשחק של השחקן הראשון.

**מהלך המשחק**

**My Ships -** מציג את הצוללות של השחקן הנוכחי, מציג היכן הן ממוקמות ואת הפגיעות אשר הן ספגו מהשחקן השני, צוללת מסומנת באייקון  , צוללת פוגעה מסומנת ב ,כמו כן, מעל ללוח בצד ימין, מופיעים המוקשים הזמינים במשחק (מסומנים באייקון ), את המוקשים הנ''ל ניתן להציב בלוח ע"י גרירת המוקש למשבצת הרצויה.

**Opponent's Ships** - מציג את לוח התקיפות של השחקן הנוכחי, תקיפות מוצלחות מסומנות באייקון  החטאות מסומנות באייקון .

**Statistics -** חלון המידע של המשחק, מציג את הסטטיסטיקות העדכניות של השחקן הנוכחי, כמה תורות שיחק, כמה פגיעות מוצלחות עשה, כמה החטאות ואת הזמן הממוצע של של תור אותו שיחק.

**History** - חלון המציג את היסטוריית המהלכים שהשחקן הנוכחי ביצע, ליד כל מהלך מצויין האם השחקן פגע או לא ואיזה נקודה ניסה לתקוף

**Remain Ships -** חלון המידע מציגה את הספינות אשר נותרו ליריב, החלון מציג את מספר הספינות שנותרו לכל יריב מכל סוג שהוגדר בתחילת המשחק.

**בונוסים שמומשו**

1. **אפשרות להחלפת "סקינים של המשחק" - ניתן לבחור בין 3 סקינים שונים של המשחק, ע''י בחירה בתפריט העליון באפשרות Style.**

**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. Players - Contains all logic and Class that related to player interface, ships, mines and all other.
   3. Exception - Contains all the exception we might throw during the game.
3. Runners - Contain the VIEW layer, so called runner, for example an Console runner will implement the game view layer using the console output. And javaFx will implement the viewr layer using JavaFx.

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.GameManagerFactory - Factory Class handle all related functionality of constructing a GameManager object, handling validation and raising exceptions.
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.players.GridEntity - implement all different kind of entity that could be placed on the board of the game
9. game.players.Mine - Implement the interface of GridEntity, and represent a Mine in the game.
10. game.players.Ship - Implement the interface of GridEntity, and represent a Ship in the game.

Enums

1. game.engine.GameState - enum for game states (Running, In Progress and Done).
2. game.engine.GameMode - enum for game mode (Basic or Advance)
3. game.engine.HitType - enum for all different types of hit (Miss, Ship hit, Mine Hit..)