משחק טאקי רב-משתמשים – תרגיל 3

מגישים : עמית טובים ת.ז. 040207847 [amit.tovim@gmail.com](mailto:amit.tovim@gmail.com)

דור אמר ת.ז. 301113403 [bndor.am@gmail.com](mailto:bndor.am@gmail.com)

הוראות הפעלה :

* פתח את הקבצים מן הכיווץ לספרית היעד .
* פתח את קובץ readme זה וקרא אותו.
* פתח את הקונסול תחת אותה ספריה
  + הרץ NPM INSTALL
  + הרץ NPM RUN BUILD
* להפעלת השרת:
  + הרץ npm run start ( או הרץ node src/server.js)
* כדי להתחבר למשחק:

פתח דפדפן ופנה לפורט 3000. (כתוב בשורת הכתובת : localhost:3010 )

הנחות יסוד במשחק טאקי

* אם שחקן מניח קלף טאקי, הוא מניח אחריו את כל הקלפים מאותו הצבע שיש לו. ואינו יכול להניח בסוף משנה צבע.
* לחיצה עם העכבר על הקלף העליון מהחפיסה – מושכת קלף מהקופה
* לחיצה על קלף מהקלפים של השחקן מניחה את אותו קלף בערימה
* לחיצה על STOP או PLUS מאפשרת אוטומטית ביצוע עוד פעולה של הנחת קלף או לקיחת קלף.
* לעבור בין מסכי המשחק בסוף כל משחק מתבצעת ע"י לחיצה על החצים למעלה.

הבונוסים שמומשו

מבנה הקוד

הפרדנו בין הקבצים השונים ע"י שמם תפקידם וכדי שיהיה קל יותר למצוא כל דבר.

רוב הקבצים דומים לצורתם בתרגיל 2 וההסבר התחתון תקף לגביהם.

Src\server.js –

Src\server\ – קובץ השרת הראשי.

enums-node – Enums של צד שרת

Users.js – מרכז את כל ניתובי השרת שמטפלים בדף הרישום למשחק  
Lobby.js – מרכז את כל ניתובי השרת שמטפלים בדף הלובי של המשחק  
game.js – מרכז את כל ניתובי השרת שמטפלים בדף המשחק עצמו

Authentication.js – מרכז את כל פעולות האימותים שרת מול לקוח.

Database.js – מכיל את ה-DB המדומה שלנו ואחראי לניהול כל המשחקים

Server-game-utils.js – מכיל את ארגז הכלים איתו database מתחזק ומנהל את כל המשחקים

Logic\ – מכיל את כל הממשקים של המשחק, וכלים לוגיים לשימוש המשחק

Src\app\serverGame\Components –

* Game.component.js היא הקומפוננטה הראשית משמכילה את לוח המשחק בתוספת Navbar עם כפתורי פעולה ומידע על המחשק, Console שמציג את התנהלות המשחק ואת הפעולות שקרו, Loader וכו'
* בתוך תיקיית app יש חלוקה ל- api-models
  + כל הממשקים של המשחק,
  + Enums של כל המשחק
  + Shared זוהי תייקיה המכילה קומפוננטות משותפות (Reusable self contained) שמהוות את אבני הבניין של המשחק
  + Game – מכיל את כל קומפוננטות המשחק עצמו בצורה היררכית, כל קומפננטה מכילה קובץ עיצוב css וקובץ js שם מוגדר הריאקט קומפוננט
* בתוך תיקיית logic מבצעת הלוגיקה של המשחק (יצירת משחק חדש, ביצוע פעולות., העברת קלפים מחבילה לחבילה וכו')
* בעצם הפרדה זאת מהווה הפרדה טובה בין הלוגיקה ל-UI, תיקייתה logic מהווה בסיס לשרת.
* התקשורת בין ה- UI ללוגיקה מתבצעת בעזרת שליחה וקבלת promises