**כללי:**

תמיכה ב-resize.

**Users - sign up**

לשמור כ -cookie את ה-session של המשתמש.

**GamesRoom -**

* לקחת ולהציג את היוזרים מ-UserManager.
* לקחת המשחקים מה-GameConfigsManager ולהציג.
* לשנות GameConfigsManager
* טעינת משחק
* להוסיף סוג משחק Advanced/Standard
* להוסיף חיווי - status לגבי האם יש שחקן שמחכה למשחק וניתן להצטרף.
* לדאוג לכך שכל שתי שניות יהיה ריפרש למצב.
* Logout - למחוק מה-UserManager. (לבדוק לגבי הבונוס)

**טעינת קובץ**

* לטעון קובץ רק אם הוא עומד בפורמט, ולא קיים קובץ כזה במערכת.
* לאפשר Join Game.
* לאפשר Join game/Watch game - לוגיקה: רק כאשר משחק כבר active , ניתן להכנס כצופה**.**

**Game -**

* סוף משחק יוצאים.
* שחקן שלא תורו נעול להכל.
* **קישור ל-GameConfigsManager**

**בעזרת**

()initiateGameFromGenerated

**8.10.17**

שינויים עיקריים:

Servlets:

**:ReadXMLServlet**

מקבל קובץ xml מUI ואז מפרסר את לקןבץ .

config, תוך כדי יוצר קבצים זמניים במחשב (העתק של הקובץ שהועלה) כדי להזין ללוגיקה.

**AddUserServlet**

מוסיף user למשחק (gameRecord).

מקבל כפרמטר PARTICIPANT/WATCHER ומוסיף למשחק בהתאם.

**Game records servlet**

מחזיר את כל ה-GameRecords ב-JSON.

**UsersServelet**

מחזיר את כל המשתמשים ב-JSON.

**Game record**

אובייקט של משחק בלוגיקה, מכיל את כל השדות הדרושים לנו.

ב-UI יופיע כ-JSON לפחות בינתיים.

ב-gameRoom.js בעיקר בונים את הטבלאות וממשים את הקריאות לשרתים.

תגיד לי אם יש משהו שלא ברור שם.

הערה: ממש כדאי לרכז את כל הקבועים שלנו ב-Constants, ושיהיו זהים בין הלוגיקה ל-UI, אחרת זה יוצר הרבה בעיות כשעוברים בין UI למודל.