**תרגיל 3 בקורס ג'אווה ואינטרנט**

**מגישים:**

**עומר בלכמן ת.ז. 205706278 – omerblechman@gmail.com**

**אסף אברהמוף ת.ז. 311305098 – assafab01@gmail.com**

**נציין כי אופן הכנסת הדיסקות לעמודה X, היא על ידי לחיצה על המשבצת הראשונה באותה עמודה. באופן דומה במצב משחק popout כאשר המשתמש רוצה למשוך דיסקית עליו ללחוץ על עמודה X בשורה התחתונה.**

**URL: http://localhost:8080/NinARow**

**הפעלת התוכנית**

**מסך כניסה** – על המשתמש להכניס שם משתמש אשר אינו ריק ולא מופיע במערכת. במידה והמשתמש הכניס קלט שאינו עונה על הדרישות הנ"ל תוצג לו הודעת שגיאה. במידה והצליח, יעבור למסך לובי.

**מסך לובי** – על המשתמש לבחור באחת מהאופציות הבאות:

1. **Choose file** – לאחר לחיצה על הכפתור יפתח למשתמש file choicer ויצטרך לטעון קובץ xml תקין שבו יהיו נתוני המשחק, במידה ונתוני המשחק לא תוצג למשתמש הודעה בהתאם לסוג השגיאה שיש בקובץ.
2. **Upload file** – לאחר לחיצה על הכפתור הקובץ שנטען למערכת יהפוך ללוח משחק. במידה ולא נטען עדיין קובץ, תוצג שגיאה למשתמש.
3. **logout** – כפתור זה ימחק את נתוני המשתמש מהמערכת ותעביר את המשתמש למסך הכניסה.
4. **Join** – לאחר טעינת משחק, תתאפשר למשתמש האפשרות להצטרף למשחק על ידי כפתור join בשורת המשחק, לאחר לחיצה על כפתור זה, ירשם המשתמש למשחק ויועבר לדף המשחק. לאחר התחלת המשחק הכפתור יהפוך להיות disable ורק לאחר סיום המשחק ויציאת כל המשתמשים ממנו, הכפתור יחזור להיות enable.
5. **view** – לאחר טעינת משחק, תתאפשר למשתמש האפשרות להצטרף למשחק בתור צופה, לאחר לחיצה על כפתור זה, יועבר המשתמש לדף המשחק אך ירשם בתור צופה.

**מסך המשחק** – בעת מעבר למסך המשחק, תוצג למשתמש הודעה לכמה אנשים מחכים על מנת להתחיל משחק. בצד שמאל של המסך יוצגו הנתונים השוטפים הקשורים למשחק + הצ'ט + הרשימת צופים. באופן דומה לתרגיל 2, בהתאם לסוג המשחק יצטרך המשתמש ללחוץ. במידה והמשחק הסתיים עקב תיקו או ניצחון, תוצג למשתמש הודעה מתאימה.

**מהלך משחק:**

1. **logout** – בעת לחיצה על כפתור זה, השחקן הנוכחי פורש, הדיסקאות שלו יעלמו מהלוח והמשחק ימשיך לשחקן הבא בתור. בנוסף יוצג בסטטיסטיקות כי השחקן פרש. במידה ונותרו שני שחקנים ושחקן אחד פרש, יוצג כי השחקן שנותר הוא המנצח של המשחק ולאחר מכן תתאפשר להתחיל משחק חדש או לטעון קובץ משחק חדש. המשתמש יעבור למסך הלובי. **בנוסף בעת סיום משחק, על המשתמש ללחוץ על כפתור זה כדי לצאת מהמשחק ולחזור למסך הלובי להצטרפות למשחק אחר.**
2. **CurrentTurnDetails** – קומפוננטה זו תציג חיווי למשתמש תור מי עכשיו, תור השחקן הבא, זמן שחלף עד כה מתחילת המשחק, סוג המשחק ומה הרצף הנדרש.
3. **HistroyMove** – קומפוננטה זו תציג למשתמש את רצף המהלכים שהיו עד כה.
4. **PlayersDetails** – קומפוננטה זו תציג למשתמש את הסטטיסטיקות של כל השחקנים המשתתפים במשחק. עבור כל שחקן יוצג מספר התורות ששיחק עד עכשיו, השם, הצבע על המגרש, סוג שחקן (אנושי או ממוחשב).

**הערה**: במידה ושחקן פרש, יעלמו נתוניו מתוכן הקומפוננטה.

1. **Chat** – בקומפוננטה זו יכולים המשתמשים להתכתב ביניהם.

**הערה**: צופים מהצד לא רואים ולא יכולים להשתתף בצ'ט המשחק.

1. **Viewers**– קומפוננטה זו תציג למשתמש את רשימת הצופים במשחק.

**סיום משחק:**

בעת סיום המשחק יוצג למשתמש הרצף המנצח עם אנימציה ושם המנצח/ים יוצג גם הוא. לאחר מכן על המשתמש ללחוץ על כפתור הlogout על מנת לחזור למסך הלובי.

**רשימת בונוסים שמומשו:**

1. **Chat** – בעת מעבר ללוח המשחק, יוצג למשתמשי המשחק (רק לשחקנים הפעילים ולא לצופים מן הצד) האפשרות להתכתב ביניהם.
2. **Viewers** – למשתמש יש אפשרות להצטרף למשחק כצופה על ידי הלחיצה על כפתור view בשורת המשחק שאליה רוצה להצטרף כצופה. הצופה יראה את התנהלות המשחק ותינתן לו האפשרות היחידה לצאת מהמשחק כצופה על ידי הלחיצה על כפתור הlogout.

**מחלקות עיקריות**

להלן המחלקות **העיקריות** בתוכנית (לפי Package):

**EngineGame**

**EngineGame** – המחלקה האחראית על הלוגיקה של המשחק. מחלקה זו מכילה בתוכה מתודות Public המציעות את כל השירותים הנחוצים למשחק – טעינת שלב, אתחול המשחק, ביצוע מהלכים, ביטול מהלכים וכיו"ב. לצורך ניהול המשחק היא שומרת בתוכה data members לצרכים הבאים:

* m\_Status – מצב המשחק הנוכחי.
* m\_Board – לוח המשחק המכיל את נתונים המשחק לצורך שמירה של הנתונים הלוגים שעודכנו עד כה.
* m\_Sequence – הרצף המבוקש למשחק.
* m\_RegisterPlayers – collection של כל השחקנים.
* m\_Turn – אינדקס של השחקן הנוכחי ב-Collection השחקנים.
* m\_StartTime – זמן תחילת המשחק (לצורך חישוב הזמן שעבר מאז תחילת המשחק).
* m\_Draw – דגל שמודלק במידה ויש תיקו.
* m\_Winner – אובייקט שמכיל את הנתונים של המנצח.
* m\_HistoryMoves – מכיל את רשימת המהלכים שבוצעו.
* m\_GameLoader – דגל שמייצג האם נטען מידע למערכת.
* m\_DataParser – אובייקט שמפענח את הנתונים של מקור המידע.
* m\_Varient – מכיל את סוג המשחק המשוחק.

**BoardGame**

מחלקה שאחראית על ניהול לוח המשחק מבחינה לוגית.

**DataHistoryDisc**

מחלקה ששומרת את הנתונים הנחוצים לצורך הצגת של מהלך מסוים שבוצע.

**PlayerEngine**

מחלקה שאחראית על שמירת הנתונים הרלוונטיים של השחקן ועדכונם לפי המהלכים שהשחקן מבצע.

**Parser:**

ממשק אשר מי שמממש אותו מפענח את הנתונים ממקור המידע.

**Boards**

**Board**

מחלקה זו אחראית לקשר בין הservlet לבין המשחק עצמו ולדאוג לעדכן את הEngine במידת הצורך ולהחזיר תשובות מהEngine שיועברו בדרך כזאת או אחרת לClient.

**BoardsManager**

מחלקה זאת מנהלת את כל המשחקים שקיימים במערכת.

**Responses**

**BoardGameResponse, HistoryContentResponse, LobbyContentResponse, PlayerDetails, PlayersDetailsResponse, UserDetailsContentResponse, ViewerListContentRespone**

מחלקות אלו אחראיות להכיל את הנתונים שיעברו parsing בעת החזרת התשובה לClient בפורמט JSON.

**Users**

**User**

מחלקה זו אחראית להכיל מידע על משתמש בודד.

**UsersManager**

מחלקה זאת מנהלת את כל המשתמשים שקיימים במערכת.

**Servlets**

**LoginServlet**

מחלקה זו מאזינה לרישום של המשתמש למערכת ודואגת להודיע למשתמש במידה וההתחברות לא הצליחה.

**LobbyServlet**

מחלקה זו מעדכנת למשתמש את מצב מסך הלובי מבחינת רשימת המשתמשים הרשומים למערכת ורשימת הלוחות שנוצרו. בנוסף במידה והClient רוצה להעלות את המשחק כלומר, ולהוסיפו לרשימת המשחק הקיימים, המחלקה תדע לטפל בכך.

**EnterGameServlet**

מחלקה זו מכניסה את המשתמש למשחק הרצוי.

**ViewerEnterServlet**

מחלקה זו מכניסה את המשתמש למשחק הרצוי בתור צופה מן הצד.

**LogoutLobbyServlet**

מחלקה זו מוציאה את המשתמש ממסך הלובי ומעבירה אותו למסך הכניסה.

**BoardGameServlet**

מחלקה זו דואגת לעדכון תוכן לוח המשחק.

**ChatServlet**

מחלקה זו דואגת לניהול תוכן קומפוננטת הצ'ט.

**PlayerMoveServlet**

מחלקה זו דואגת להעברת המידע ועדכונו במנוע המשחק בעת ביצוע מהלך של שחקן אנושי.

**ComputerMoveServlet**

מחלקה זו דואגת לביצוע תור של שחקן ממוחשב.

**LogoutGameServlet**

מחלקה זו מוציאה את השחקן ממסך המשחק, מעבירה אותו למסך הלובי ומוחקת את נתוניו מתוכן המשחק וגם את שם המשחק מsession של אותו שחקן.

**HistoryServlet**

המחלקה אחראית על מהלכי היסטוריית המשחק.

**PlayersDetailsServlet**

מחלקה זו אחראית על עדכון נתוני השחקנים הפעילים.

**StartGameServlet**

מחלקה זו אחראית להתחיל את המשחק.

**StatisticsServlet**

מחלקה זו אחראית על עדכון הסטטיסטיקות של המשחק.

**ViewersListServlet**

מחלקה זו אחראית לעדכן את רשימת הצופים במשחק.

**UI**

**LoginModel**

אחראית על התצוגה של מסך הכניסה.

**LobbyArea**

אחראית על התצוגה של מסך הלובי.

**GameRoom**

אחראית על התצוגה של מסך המשחק.